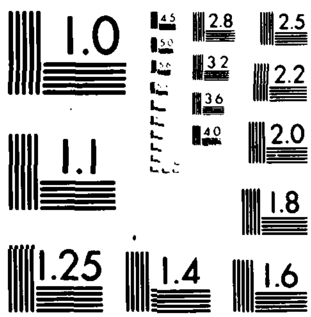
AD-A086 695 CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING A--ETC F/G 17/2
COLLISION-RESOLUTION ALGORITHMS AND RANDOM-ACCESS COMMUNICATION--ETC(U)
APR 80 J L MASSEY N00014-78-C-0778
UNCLASSIFIED UCLA-ENG-8016 NL

END
DATE
FILMED
8-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

UCLA-ENG-8016
APRIL 1980

COLLISION-RESOLUTION ALGORITHMS AND RANDOM-ACCESS COMMUNICATIONS

JAMES L. MASSEY

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 UCLA-ENG-8016	2. GOVT ACCESSION NO. AD-A086695	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 COLLISION-RESOLUTION ALGORITHMS AND RANDOM-ACCESS COMMUNICATIONS.		5. TYPE OF REPORT & PERIOD COVERED 9 Interim rept.
		6. PERFORMING ORG. REPORT NUMBER UCLA-ENG-8016
7. AUTHOR(s) 10 James L. Massey		8. CONTRACT OR GRANT NUMBER(s) 15 ONR N00014-78-C-0778 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California School of Engineering and Applied Science Los Angeles, CA 90024		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Branch Office 1030 East Green Street Pasadena, CA 91101	11	12. REPORT DATE Apr[redacted]80 13. NUMBER OF PAGES 74+iii
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Research Laboratory Code 7557 Washington, DC 20375	12 80	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Random-access, Aloha system, multi-access, delay-throughput, conflict resolution, collision-resolution

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The problem considered is the random-accessing by very many transmitters of a common receiver over a time-slotted collision-type channel with feedback. A collision-resolution algorithm (CRA) is a protocol for the transmission and retransmission of packets such that, after a collision, all transmitters eventually and simultaneously learn that all the colliding packets have been successfully retransmitted. Focus is placed on a CRA due to Capetanakis (the CCRA) and a slight modification thereof (the MCCRA).

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

404637 JM

The fundamental (traffic-independent) properties of the CCRA and MCCRA are derived by means of a recursive analysis, with emphasis on the mean and variance of the number of slots required to resolve a collision among N packets. These results are then used to analyze by a Markovian approach the performance of various random-access algorithms, built upon the CCRA and MCCRA, for the case of infinitely many identical sources generating Poisson traffic. The maximum stable throughput is determined for each random-access algorithm, and tight upper and lower bounds are developed for the delay-throughput characteristic of the "obvious" random-access algorithm built upon the CCRA.

The previous analyses assumed delayless propagation, noiseless feedback, and a forward channel that was noiseless except for collisions. The ideal case analysis is extended to incorporate propagation delays and channel errors. The CCRA is shown to be impervious to such errors, but the MCCRA is shown to be extremely sensitive to channel errors. The use of carrier-sensing to increase the maximum stable throughput is analyzed.

Throughout this report, the analysis is mathematically rigorous and makes no appeal to the hypothesis of "statistical equilibrium" that has characterized most studies of random-access systems.

COLLISION-RESOLUTION ALGORITHMS AND RANDOM-ACCESS COMMUNICATIONS

James L. Massey

School of Engineering and Applied Science
University of California
Los Angeles, California 90024

[This report is a preprint of a paper to appear in the contributed volume, Multi-User Communications (Ed. G. Longo), to be published by Springer-Verlag in the CISM Courses and Lectures series.]

This work was supported by the
U.S. Office of Naval Research
under Contract No. N00014-78-C-0778

ABSTRACT

The problem considered is the random-accessing by very many transmitters of a common receiver over a time-slotted collision-type channel with feedback. A collision-resolution algorithm (CRA) is a protocol for the transmission and retransmission of packets such that, after a collision, all transmitters eventually and simultaneously learn that all the colliding packets have been successfully retransmitted. Focus is placed on a CRA due to Capetanakis (the CCRA) and a slight modification thereof (the MCCRA).

The fundamental (traffic-independent) properties of the CCRA and MCCRA are derived by means of a recursive analysis, with emphasis on the mean and variance of the number of slots required to resolve a collision among N packets. These results are then used to analyze by a Markovian approach the performance of various random-access algorithms, built upon the CCRA and MCCRA, for the case of infinitely many identical sources generating Poisson traffic. The maximum stable throughput is determined for each random-access algorithm, and tight upper and lower bounds are developed for the delay-throughput characteristic of the "obvious" random-access algorithm built upon the CCRA.

The previous analyses assumed delayless propagation, noiseless feedback, and a forward channel that was noiseless except for collisions. This ideal case analysis is extended to incorporate propagation delays and channel errors. The CCRA is shown to be impervious to such errors, but the MCCRA is shown to be extremely sensitive to channel errors. The use of carrier-sensing to increase the maximum stable throughput is analyzed.

Throughout this report, the analysis is mathematically rigorous and makes no appeal to the hypothesis of "statistical equilibrium" that has characterized most studies of random-access systems.

TABLE OF CONTENTS

Page No.

1. INTRODUCTION

Communications engineers have a long acquaintance with the "multiple-access" problem, i.e., the problem of providing the means whereby many senders of information can share a common communications resource. The "classical" solution has been to do some form of multiplexing (e.g., time-division multiplexing (TDM) or frequency-division multiplexing (FDM) in order to parcel out the resource equitably among the senders.) A fixed division of the resources, however, becomes inefficient when the requirements of the users vary with time. The classical "fix" for the multiplexing solution is to add some form of demand-assignment so that the particular division of resources can be adapted to meet the changing requirements. Such demand-assigned multiplexing techniques have proved their worth in a myriad of multiple-access applications.

A second solution of the multiple-access problem is to employ some form of random-access, i.e., to permit any sender to seize the entire communications resource when he happens to have information to transmit. The random-access solution is actually older than the multiplexing solution. For instance, the technique by which "ham" operators share a particular radio frequency channel is a random-access one. If two hams come up on the channel at virtually the same time, their transmissions interfere. But the inherent randomness in human actions ensures that eventually one will repeat his call well enough in advance of the other that the latter hears the former's signal and remains quiet, allowing the former to seize the channel. Moreover, essentially the same random-access technique is used by many people around the same table to communicate with one another over the same acoustical channel.

A better name for the time-division multiplexing (either with or without demand assignment) solution to the multi-access problem might be scheduled-access. Any sender knows that eventually he will be granted sole access to

the channel, perhaps to send some information or perhaps to ask for a larger share of the resources. The key consequence is that the resources will be wasted during the period that he is granted sole access when in fact he has nothing to say. Thus, scheduled-access techniques tend to become inefficient when there are a large number of senders, each of which has nothing to say most of the time. But this is just the situation where random-access techniques tend to become efficient.

The computer age has given rise to many multiple-access situations in which there are a large number of senders, each of which has nothing to say most of the time. One such situation, namely the problem of communicating from remote terminals on various islands of Hawaii via a common radio channel to the main computer, led to the invention by Abramson [1] of the first formal random-access algorithm, now commonly called pure Aloha. Here it is supposed that transmitters can send data only in packets of some fixed duration, say T seconds. In pure Aloha, a transmitter always transmits such a packet at the moment it is presented to the transmitter by its associated information source. If no other transmitter is active during this T second transmission, then the packet is successfully received. Otherwise, there is a "collision" that is assumed to destroy all the packets that overlap. It is further supposed that, via some form of feedback, the transmitters discover whether or not their packets suffer collisions. When a collision occurs, the packets must be retransmitted. To avoid a repetition of the same collision, pure Aloha specifies that after a collision each transmitter involved randomly selects a waiting time before it again retransmits its packet. Assuming a Poisson traffic model and "statistical equilibrium," Abramson showed that pure Aloha had a maximum "throughput" of $\frac{1}{2e} \approx .184$, computed as the fraction of time on the channel occupied by successfully transmitted packets. It was soon noticed by Roberts [2] that the maximum

throughput could be doubled to $\frac{1}{e} \approx .368$ by "slotting time" into T second intervals and requiring that the transmission of a packet be slightly delayed (if necessary) to coincide with a slot. This modification to Abramson's algorithm is now known as the slotted-Aloha random-access algorithm.

In his 1970 paper that first proposed the pure Aloha algorithm, Abramson introduced the hypothesis of "statistical equilibrium" in order to analyze the algorithm's performance for a Poisson traffic model. Essentially, this hypothesis states that the algorithm will eventually reach a steady-state situation in which the traffic from retransmitting of messages will form a stationary Poisson process that is independent of the new message traffic. It is precisely this assumption that leads to the maximum throughput bounds of 1/2e and 1/e for pure Aloha and slotted-Aloha, respectively. Abramson's statistical equilibrium assumption was a bold one and was best justified by the fact that, without it, the analytical tools appropriate for treating his algorithm did not exist. As time went on, however, communications engineers generally forgot that there was neither mathematical nor experimental justification for this hypothesis of "statistical equilibrium," and came to accept the numbers 1/2e and 1/e as the "capacities" of the pure Aloha channel and slotted Aloha channel, respectively, for the Poisson traffic model. Even more unfortunately, most workers continued to invoke the hypothesis of "statistical equilibrium" to "prove" that their particular rococo extension of the Aloha algorithm had superior delay-throughput properties compared to all previous ones, even though the character of their refinements should have made the hypothesis all the more suspect.

The next breath of truly fresh air in the research on random-access algorithms came in a 1977 M.I.T. doctoral dissertation by Capetanakis [3]. Capetanakis departed from the path beaten by Abramson in two important ways. First, he showed how, without prior scheduling or central control, the

transmitters with packets to retransmit could make use of the known past history of collisions to cooperate in getting those packets through the channel. Second, he eliminated statistical equilibrium as an hypothesis by proving mathematically that his algorithm would reach a steady-state, albeit a highly non-Poisson one for the retransmitted traffic, when the new packet process was Poisson with a rate less than some specific limit. It must have come as a bombshell to many that Capetanakis could prove that his scheme achieves throughputs above the 1/e "barrier" for slotted-Aloha.

The aim of this paper is to illuminate the key features of Capetanakis' work and the subsequent work by others based on it, and to expose some analytical methods that appear useful in such studies. We also introduce a few new results of our own.

In Section 2 we formulate the concept of a "collision-resolution algorithm" and treat Capetanakis' algorithm within that context. In Section 3, we catalog those properties of the Capetanakis collision-resolution algorithm that are independent of the random process describing the generation of new packets. Then, in Section 4, we use these properties to analyze the performance of the Capetanakis random-access algorithm (and its variants) for the Poisson traffic model under idealized conditions. In Section 5, we study quantitatively the effects of relaxing the idealized assumptions to admit propagation delays, channel errors, etc. Finally, we summarize the main conclusions of our study and review the historical development of the central concepts, hopefully with due credit to the original contributors.

2. COLLISION-RESOLUTION ALGORITHMS

2.1 General Assumptions

We wish to consider the random-accessing by many transmitters of a common receiver under the following idealized conditions:

(i) The forward channel to the receiver is a time-slotted collision-type channel, but is otherwise noiseless. The transmitters can transmit only in "packets" whose duration is one slot. A "collision" between two or more packets is always detected as such at the receiver, but the individual packets cannot be reconstructed at the receiver.

(ii) The feedback channel from the common receiver is a noiseless broadcast channel that informs the transmitters immediately at the end of each slot whether (a) that slot was empty, or (b) that slot contained one packet (which was thus successfully transmitted), or (c) that slot contained a collision of two or more packets (which must thus be retransmitted at later times.)

(iii) Propagation delays are negligible, so that the feedback information for slot i can be used to determine who should transmit in the following slot.

In later sections, we shall relax each of these conditions to obtain a more realistic model of a random access system. We shall see, however, that the analysis for the idealized case can be readily generalized to incorporate more realistic assumptions.

2.2 Definition of a Collision-Resolution Algorithm

By a collision-resolution algorithm for the random-accessing of a collision-type channel with feedback, we mean a protocol for the transmission and retransmission of packets by the individual transmitters with the property that after each collision all packets involved in the collision are eventually retransmitted successfully and all transmitters (not only those whose packets collided) eventually and simultaneously become aware that these packets have

been successfully retransmitted. We will say that the collision is resolved precisely at the point where all the transmitters simultaneously become aware that the colliding packets have all been successfully retransmitted.

It is not at all obvious that collision-resolution algorithms exist. The Aloha algorithm, for instance, is not a collision-resolution algorithm as one can never be sure that all the packets involved in any collision have been successfully transmitted. Thus, the recent discovery by Capetanakis [3-5] of a collision-resolution algorithm was a surprising development in the evolution of random-access techniques whose full impact has yet to be felt.

It might seem that a collision-resolution algorithm would require a freeze after a collision on the transmission of new packets until the collision had been resolved. In fact, Capetankis' algorithm does impose such a freeze. However, as we shall see later, one can devise collision-resolution algorithms that incorporate no freeze on new packet transmissions — another somewhat surprising fact.

2.3 The Capetanakis Collision-Resolution Algorithm (CCRA)

In [3,4], Capetanakis introduced the collision-resolution algorithm of central interest in this paper and which he called the "serial tree algorithm." We shall refer to this algorithm as the Capetanakis collision-resolution algorithm (CCRA). The CCRA can be stated as follows:

> CCRA: After a collision, all transmitters involved flip a binary fair coin; those flipping 0 retransmit in the very next slot, those flipping 1 retransmit in the next slot after the collision (if any) among those flipping 0 has been resolved. No new packets may be transmitted until after the initial collision has been resolved.

The following example should both clarify the algorithm and illustrate its main features.

Suppose that the initial collision is among 4 transmitters, as shown in

Figure 2.1. For convenience, we refer to these transmitters as A, B, C and D.

After the collision in slot 1, all four of these transmitters flip their binary coins — we suppose that B and C flip 0 while A and D flip 1. Thus B and C flip again at the end of slot 2 — we suppose that C flips 0 while B flips 1. Thus, only C sends in slot 3 and his packet is now successfully transmitted. B thus recognizes that he should send in slot 4, and his packet is now successfully transmitted.

It is illuminating to study the action thus far in the algorithm on the tree diagram in Figure 2.2 in which the number above each node indicates the time slot, the number inside the node indicates the feedback information for that time slot (0 = empty slot, 1 = single packet, ≥ 2 = collision), and the binary numbers on the branches coming from a node indicate the path followed by these transmitters that flipped that binary number after the collision at that node. Thus, <u>collisions correspond to intermediate nodes</u> in this binary rooted tree since such nodes must be extended by the algorithm before the collision at that node is resolved. On the other hand, <u>empty slots or slots with only one packet correspond to terminal nodes</u> in this binary rooted tree because after the corresponding transmission <u>all</u> transmitters simultaneously learn that any transmitter sending in that slot (i.e., zero or one transmitters) has successfully transmitted his message. Thus, a collision is resolved when and only when the algorithm has advanced to the point that the corresponding node forms the root node of a completed binary subtree. Thus, from Figure 2.2, which illustrates the same situation as Figure 2.1, we see that the collision in slot 2 is resolved in slot 4. Thus, transmitters A and D, who have been patiently waiting since the collision in slot 1, recognize that they should now retransmit in slot 5.

After the collision in slot 5, we suppose that A and D both flip 1. Thus,

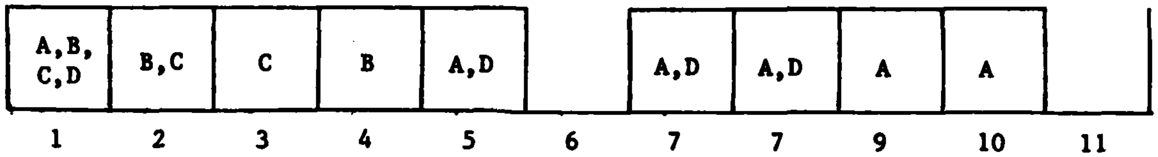


Fig. 2.1: Example of a Collision Resolution Interval for the CCRA.

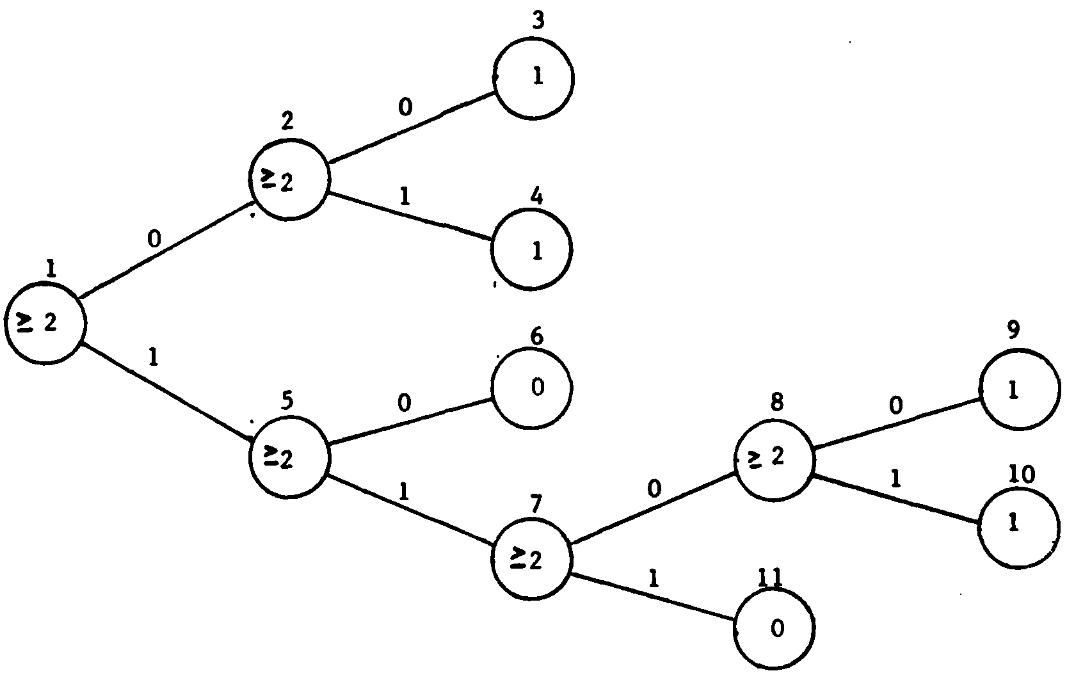


Fig. 2.2: Tree Diagram for the Collision Resolution Interval of Fig. 2.1.

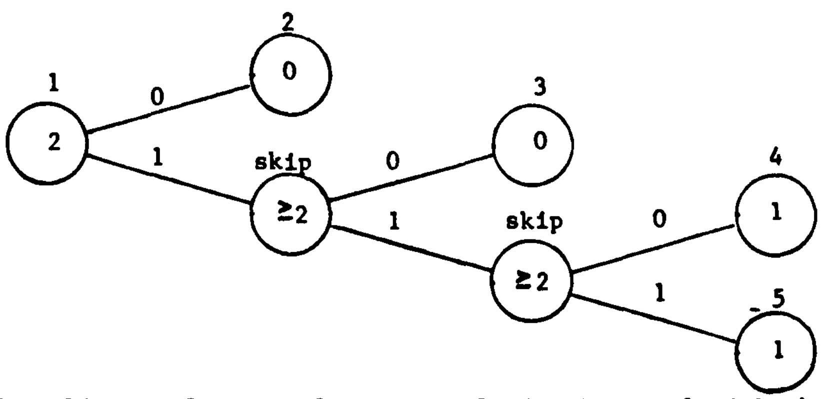


Fig. 2.3: Tree Diagram for a Collision Resolution Interval with the MCCRA.

slot 6 is empty, so A and D again recognize that they should retransmit in slot 7. After the collision in slot 7, we suppose that A and D both flip 0. Hence they both retransmit in slot 8. After the collision in slot 8, we suppose that A flips 0 and D flips 1. Thus, A successfully transmits in slot 9 and D successfully transmits in slot 10. All four transmitters in the original collision have now transmitted successfully, but the collision is not yet resolved. The reason is that no one can be sure that there was not another transmitter, say E, who transmitted in slot 1 then flipped 1 and who retransmitted in slot 5 and then flipped 1 and who retransmitted again in slot 7 and then flipped 1 and who thus is now waiting to retransmit in slot 11. It is not until slot 11 proves to be empty that the original collision is finally resolved. All transmitters (not just the four in the original collision) can grow the tree in Figure 2.2 from the feedback information, and thus all transmitters now simultaneously learn that all the packets in the original collision have now been successfully transmitted.

Because a binary rooted tree (i.e., a tree in which either 2 or no branches extend from the root and from each subsequent node) has exactly one more terminal node than it has intermediate nodes (or two more terminal nodes then intermediate nodes excluding the root node), we have the following:

> <u>CCRA Property</u>: A collision in some slot is resolved precisely when the number of subsequent collision-free slots exceeds by two the number of subsequent slots with collisions.

For instance, from Figure 2.1 we see that the collision in slot 2 is resolved in slot 4, the collision in slot 5 is resolved in slot 11, the collision in slot 1 is also resolved in slot 11, etc. Notice that the later collisions are resolved sooner.

The above CCRA Property suggests a simple way to implement the CCRA due to R. G. Gallager [6].

CCRA Implementation: When a transmitter flips 1 following a collision in which he is involved, he sets a counter to 1, then increments it by one for each subsequent collision slot and decrements it by one for each subsequent collision-free slot. When the counter reaches 0, the transmitter retransmits in the next slot.

Additionally, all transmitters must know when the original collision (if any) has been resolved as this determines when new packets may be sent. For this purpose, it suffices for each transmitter to have a second counter which is set to 1 just prior to the first slot, then incremented by one for each subsequent collision slot and decremented by one for each subsequent collision-free slot. When this second counter reaches 0, the original collision (if any) has been resolved.

We shall refer to the period beginning with the slot containing the original collision (if any) and ending with the slot in which the original collision is resolved (or ending with the first slot when that slot is collision-free) as the collision-resolution interval (CRI). Our main interest will be in the statistics of the length of the CRI, i.e., of the number of slots in the CRI. Note that the CRI illustrated by Figure 2.1 (or, equivalently, by Figure 2.2) has length 11 slots. Since 4 packets are successfully transmitted in this CRI, its "throughput" is $4/11 \approx .364$ packets/slot. We shall soon see that this is quite typical for the throughput of a CRI with 4 packets in the first slot when the CCRA is used to resolve the collision.

It should also be noted that, in the implementation of the CCRA, the feedback information is used only to determine whether the corresponding slot has a collision or is collision-free. It is not necessary, therefore, to have the feedback information distinguish empty slots from slots with one packet when the CCRA is used.

2.4 <u>The Modified Capetanakis Collision-Resolution Algorithm</u> (See also [7].)

Referring to the example in Figure 2.2, we see that after slot 6 proves to be empty following the collision in slot 5, all transmitters now know that all the packets which collided in slot 5 will be retransmitted in slot 7. Thus, all transmitters know in advance that slot 7 will contain a collision. (Note that this statement will only be true when the feedback information distinguishes empty slots from slots with one packet.) Thus, it is wasteful actually to retransmit these packets in slot 7. The transmitters can "pretend" that this collision has taken place and immediately flip their binary coins and continue with the CCRA. The suggestion to eliminate these "certain-to-contain-a-collision" slots in the CCRA is due to the author. We shall refer to the corresponding algorithm as the modified Capetanakis collision-resolution algorithm (MCCRA). The MCCRA may be stated as follows:

> <u>MCCRA</u>: Same as the CCRA algorithm <u>except</u> that when the feedback indicates that a *slot in which* a set of transmitters who flipped 0 should retransmit is in fact empty, then each transmitter involved in the most recent collision flips a binary fair coin, those flipping 0 retransmit in the very next slot, those flipping 1 retransmit in the next slot after the collision (if any) among those flipping 0 is resolved (subject to the exception above.)

Figure 2.3 gives the binary tree for a CRI containing two packets in the original collision and for which both transmitters flipped 1 on the first two tosses of their binary coin, but one flipped a 0 and the other a 1 on their third toss. The nodes labelled "skip" and having no slot number written above them correspond to points where the feedback indicates that certain transmitters should immediately flip their binary coins to thwart a certain collision. Note that this CRI has length 5, but would have had length 7 if the unmodified CCRA had been used because then the nodes labelled "skip" would become collision slots.

We observe next that, in the MCCRA, <u>an empty slot corresponding to retransmissions by a set of transmitters who flipped 0 is precisely the same as an empty slot that is separated from the most recent collision only by empty slots</u>. This follows from the facts that transmitters who flip 0 always send in the slot immediately after their flip is made, and that a flip is made only after a collision or after a "skipped collision," i.e., after an empty slot corresponding to retransmissions by a set of transmitters who had flipped 0. This observation justifies the following:

> <u>MCCRA Implementation</u>: Each transmitter has a flag F that is initially 0 and that he sets to 1 after a collision slot and sets to 0 after a slot with one packet. When a transmitter flips 1 following a collision in which he is involved, he sets a counter to 1, then increments it by one for each subsequent collision, decrements it by 1 for each subsequent slot with one pocket, and also decrements it by one for each subsequent empty slot that occurs with F = 0. If his counter is 1 after an empty slot that occurs with F = 1 he flips his binary coin and decrements his counter by one if and only if 1 was flipped. When the counter reaches 0, the transmitter retransmits in the next slot.

Again the transmitters can use a <u>second counter</u> in conjunction with the same flag to determine when the CRI is complete. This second counter is set to 1 prior to the first slot, then incremented by one for each subsequent collision slot, decremented by one for each subsequent slot with one packet, and also decremented by one for each subsequent empty slot that occurs with F = 0. When this second counter reaches 0, the original collision (if any) has been resolved.

Because the MCCRA is merely the CCRA modified to eliminate slots where collisions are certain to occur, it will always perform at least as well as the latter algorithm. In the MCCRA, we appear to have "gotten something for nothing." This is not quite true, however, for two minor reasons:

(i) The MCCRA, unlike the CCRA, requires the feedback information to distinguish between empty slots and slots with one packet.

(ii) The MCCRA is slightly more complex to implement than the CCRA because of the necessity for the "flag" in the former algorithm.

But there is a third and far stronger reason for hesitation in preferring the MCCRA over the CCRA that we will subsequently demonstrate but that is not apparent under the idealized conditions considered in this section, namely:

(iii) When channel errors can occur, the MCCRA can suffer <u>deadlock</u>, i.e., reach a situation where the CRI never terminates and no packets are ever transmitted after some point.

3. TRAFFIC-INDEPENDENT PROPERTIES OF THE CAPETANAKIS COLLISION-RESOLUTION ALGORITHM

3.1 Definitions

In both the Capetanakis collision-resolution algorithm (CCRA) and its modification, when there is a collision in the first slot of a collision-resolution interval (CRI) no new packets may be transmitted until the CRI is completed. We shall let X denote the number of packets transmitted in the first slot of some CRI, and let Y denote the length (in slots) of this same CRI. Given X, Y depends only on the results of the coin tosses performed internally in the algorithms, and hence is independent of the traffic statistics that led to the given value of X. We can thus refer to any statistic of the CRI conditioned upon the value of X as traffic-independent. In this section, we shall study those traffic-independent properties of the CCRA and the MCCRA that are of greatest importance for the performance of random-access algorithms that incorporate these algorithms. The first and most important of these is the conditional mean CRI length, L_N, defined as

$$L_N = E(Y|X=N). \tag{3.1}$$

The conditional second moment of CRI length, S_N, defined by

$$S_N = E(Y^2|X=N), \tag{3.2}$$

is also of fundamental importance. The conditional variance of CRI length, V_N, defined by

$$V_N = Var(Y|X=N), \tag{3.3}$$

will prove to be more tractable than S_N directly, but of course is related to S_N by

$$V_N = S_N - (L_N)^2. \tag{3.4}$$

We now investigate these traffic-independent quantities in detail for the CCRA.

3.2 Intuitive Analysis

Our aim here is to determine the coarse dependence of L_N, V_N and S_N on N for the CCRA as a guide to a subsequent precise analysis. To do this, we suppose that $N = 2n$ is very large. Then, as shown in Figure 3.1, there will be a collision in slot 1 of the CRI, following which very close to half of the transmitters will flip 0 and half will flip 1. Thus

$$L_{2n} \approx 1 + 2L_n, \qquad n >> 1 \tag{3.5}$$

since the expected number of slots needed to resolve the collision in slot 2 of the approximately n transmitters who flipped 0 is L_n, following which the expected number of slots to resolve the subsequent collision of the approximately n transmitters who flipped 1 is also L_n. Considered as an equality, (3.5) is a recursion whose solution is $L_N = \alpha N - 1$ for an arbitrary constant α. Thus, we conclude that

$$L_N \approx \alpha N - 1, \qquad N >> 1 \tag{3.6}$$

describes the coarse dependence of L_N on N, which whets our appetite to find the constant α. In fact, we will soon see that (3.6) is remarkably accurate even for small values of N.

From Figure 2.1 and the fact that the number of slots needed to resolve the collision among the n transmitters who flipped 0 is independent of the number of slots needed to resolve the collision among the n transmitters who flipped 1, we see that

$$V_{2n} \approx 2V_n, \qquad n >> 1. \tag{3.7}$$

This recursion forces the conclusion that

$$V_N \approx \beta N, \qquad N >> 1 \tag{3.8}$$

for some constant β. We shall soon see that (3.8) likewise is quite accurate even for rather small N. Finally, (3.4), (3.6) and (3.8) imply

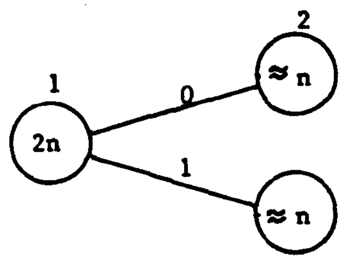


Fig. 3.1: Typical Action of the CCRA in First Slot of the CRI When n Is Large.

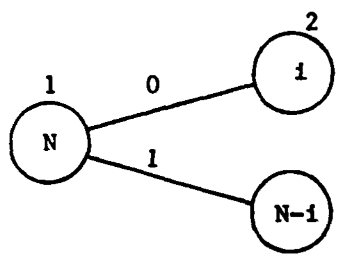


Fig. 3.2: General Action of the CCRA in First Slot of CRI when N Exceeds 1.

$$S_N \approx \alpha^2 N^2 - (2\alpha-\beta)N + 1, \qquad N >> 1, \tag{3.9}$$

which completes our intuitive analysis.

3.3 Conditional Mean of CRI Length

We now give for the CCRA a precise analysis of the expected CRI length, L_N, given that $X = N$ packets are transmitted in the first slot. When N is 0 or 1, the CRI ends with its first slot so that

$$L_0 = L_1 = 1. \tag{3.10}$$

When $N \geq 2$, there is a collision in the first slot. The probability that exactly i of the colliding transmitters flip 0 (as depicted in Figure 3.2) is just

$$p_N(i) = \binom{N}{i} 2^{-N}, \tag{3.11}$$

in which case the expected CRI length is just $1 + L_i + L_{N-i}$. Hence, it follows that

$$\begin{aligned} L_N &= 1 + \sum_{i=0}^{N} (L_i + L_{N-i}) p_N(i) \\ &= 1 + 2 \sum_{i=0}^{N} L_i p_N(i) \end{aligned} \tag{3.12}$$

where we have used the fact that $p_N(i) = p_N(N-i)$. Solving for L_N gives the recursion

$$L_N = [1 + 2 \sum_{i=0}^{N-1} L_i p_N(i)]/(1-2^{-N+1}) \tag{3.13}$$

which holds for all $N \geq 2$. The required initial conditions are given in (3.10). In Table 3.1, we give the first few values of L_N as found from (3.13).

N	L_N
0	1
1	1
2	5
3	$23/3 \approx 7.667$
4	$221/21 \approx 10.524$
5	13.419
6	16.313

Table 3.1 Expected CRI length for the CCRA given N packets in the first slot.

From Table 3.1, we can see that $L_N - L_{N-1} \approx 2.9$ for $N \geq 4$. This suggests that the constant α in (3.6) is about 2.9. In fact, we see from Table 3.1 that the $N >> 1$ approximation $L_N \approx 2.9\ \alpha - 1$ is already quite accurate for $N \geq 3$.

We now develop a technique for obtaining arbitrarily tight upper and lower bounds on L_N. We begin by choosing an arbitrary positive integer M. We seek to find a constant, α_{uM}, as small as possible, for which we can prove

$$L_N \leq \alpha_{uM} N - 1 \quad , \quad \text{all } N \geq M. \tag{3.14}$$

(The first subscript on α is only to remind us that this constant appears in an <u>upper</u> bound on L_N.) By using the Kronecker delta δ_{ij} defined to be 1 if $i = j$ and 0 if $i \neq j$, we can rewrite (3.14) as

$$L_N \leq \alpha_{uM} N - 1 + \sum_{i=0}^{M-1} \delta_{iN}(L_N - \alpha_{uM} N + 1), \text{ all } N \tag{3.15}$$

because the right side reduces to L_N for $N < M$. Substituting the bound (3.15) for those L_i on the right in (3.13) and making use of the fact that

$$\sum_{i=0}^{N-1} i p_N(i) = \sum_{i=0}^{N} i p_N(i) - N p_N(N)$$

$$= \frac{N}{2}\left(1-2^{-N+1}\right),$$

we obtain

$$L_N \leq \alpha_{uM} N - 1 + 2\left[\sum_{i=0}^{M-1} (L_i - \alpha_{uM} i + 1) p_N(i)\right] / (1-2^{-N+1}). \tag{3.16}$$

It thus follows by induction that (3.14) holds for any α_{uM} such that the summation in square brackets on the right in (3.16) is nonpositive for all $N \geq M$, i.e., such that

$$\alpha_{uM} \sum_{i=0}^{M-1} i p_N(i) \geq \sum_{i=0}^{M-1} (L_i + 1) p_N(i), \quad N \geq M. \tag{3.17}$$

The best upper bound is obtained by choosing α_{uM} such that (3.17) holds with equality in the "worst case," i.e., choosing

$$\alpha_{uM} = \sup_{N \geq M} \left[\sum_{i=0}^{M-1} \binom{N}{i} (L_i + 1) / \sum_{i=0}^{M-1} \binom{N}{i}\right] i. \tag{3.18}$$

By an entirely analogous argument, one can show inductively that

$$L_N \geq \alpha_{\ell M} N - 1 \ , \quad \text{all } N \geq M$$

holds for the choice

$$\alpha_{\ell M} = \inf_{N \geq M} \left[\sum_{i=0}^{M-1} \binom{N}{i} (L_i + 1) \Big/ \sum_{i=0}^{M-1} \binom{N}{i} i \right]. \tag{3.19}$$

For a given M, after one has calculated L_i for all $i < M$, it is a simple matter numerically to find the maximizing N and minimizing N in (3.18) and (3.19), respectively, and hence to determine α_{uM} and $\alpha_{\ell M}$. Table 3.2 summarizes these calculations and furnishes bounds on the "true coefficient" α in (3.6). From the M = 5 case, we see that

$$2.8810 \leq \alpha \leq 2.8867 \tag{3.20}$$

so that in fact we know the value of α to three significant decimal digits. We can also summarize the M = 5 results as

$$2.8810N - 1 \leq L_N \leq 2.8867N - 1 \quad , \quad N \geq 4. \tag{3.21}$$

(The inductive argument for M = 5 guarantees that (3.21) holds for $N \geq 5$; by checking against the values in Table 3.1, we find that (3.21) holds also for N = 4.) For all practical purposes, the bounds in (3.21) are so tight as to be tantamount to equalities, and we conclude that we now have determined L_N for the CCRA and are ready to move on to consider the conditional second moment of epoch length. Before doing so, it is interesting to note (as was pointed out to us by W. Sandrin of the Comsat Laboratories) that

$$\frac{2}{\ell n 2} = 2.8854 \tag{3.22}$$

which, together with the binary nature of the CCRA, lends support to the conjecture that

$$\alpha = \lim_{M \to \infty} \alpha_{uM} = \lim_{M \to \infty} \alpha_{\ell M} = \frac{2}{\ell n 2}. \tag{3.23}$$

M	α_{uM}	Maximizing N in (3.18)	$\alpha_{\ell M}$	Minimizing N in (3.19)
2	3	2	2	∞
3	3	∞	2.8750	4
4	2.8965	14	2.8810	4
5	2.8867	8	2.8810	∞

Table 3.2 Values of the coefficients α_{uM} and $\alpha_{\ell M}$ in the bounds (3.18) and (3.19), respectively, for L_N.

3.4 Conditional Variance and Second Moment of CRI Length

We now seek tight bounds on the conditional variance, $V_N = \mathrm{Var}(Y|X=N)$, and the conditional second moment, $S_N = E(Y^2|X=N)$, of the CRI length for the CCRA. Letting X_0 denote the number of transmitters who flip 0 after the collision in slot 1, we see from Figure 3.2 that

$$
\begin{aligned}
S_N &= \sum_{i=0}^{N} E(Y^2|X=N,X_0=i)p_N(i) \\
&= \sum_{i=0}^{N} [\mathrm{Var}(Y|X=N,X_0=i) + E^2(Y|X=N,X_0=i)]p_N(i) \\
&= \sum_{i=0}^{N} [V_i+V_{N-i}+(1+L_i+L_{N-i})^2]p_N(i) \\
&= \sum_{i=0}^{N} [2V_i+(1+L_i+L_{N-i})^2]p_N(i) \qquad (3.24)
\end{aligned}
$$

for $N \geq 2$, where the next-to-last equality follows from the fact that the number of slots used to resolve the collision (if any) among the i transmitters who flipped 0 is independent of the number used to resolve the collision (if any) among the N - i transmitters who flipped 1, and the last equality follows from the fact that $p_N(i) = p_N(N-i)$. Combining (3.4) and (3.24), we obtain

$$V_N = \left[2 \sum_{i=0}^{N-1} V_i p_N(i) + \sum_{i=0}^{N} (1+L_i+L_{N-i})^2 p_N(i) - L_N^2\right] \Big/ (1-2^{-N+1}) \tag{3.25}$$

for $N \geq 2$, which is our desired recursion for V_N. Because $Y = 1$ when $X = 0$ or $X = 1$, the appropriate initial conditions are

$$V_0 = V_1 = 0. \tag{3.26}$$

Using the values of L_N given in Table 3.1, we can use (3.25) to find the values of V_N given in Table 3.3, to which we have added the corresponding values of S_N found using (3.4). From Table 3.3, we see that the linear growth of V_N with N predicted asymptotically by (3.8) is already evident for $N \geq 4$, the constant of proportionality being $\beta \approx 3.4$.

N	V_N	S_N
0	0	1
1	0	1
2	8	33
3	88/9≈9.78	68.56
4	13.53	124.2
5	16.93	197.0
6	20.32	286.3

Table 3.3 Variance and second moment of CRI length for the CCRA given N packets in the first slot.

We can develop a simple lower bound on V_N by noting that, because x^2 is a convex function, Jensen's inequality [8] together with (3.12) implies

$$\sum_{i=0}^{N} (1+L_i+L_{N-i})^2 p_N(i) \geq L_N^2. \tag{3.27}$$

Substituting (3.27) in (3.25) gives the simple inequality

$$V_N \geq 2 \sum_{i=0}^{N-1} V_i P_N(i)/(1-2^{-N+1}). \tag{3.28}$$

In the same manner as led to (3.19), we can use (3.28) to verify that

$$V_N \geq \beta_{\ell M} N \quad , \qquad N \geq M \tag{3.29}$$

holds for the choice

$$\beta_{\ell M} = \inf_{N \geq M} \left[\sum_{i=0}^{M-1} \binom{N}{i} V_i \Big/ \sum_{i=0}^{M-1} \binom{N}{i} i \right]. \tag{3.30}$$

Table 3.4 gives the values of values of $\beta_{\ell M}$ for $3 \leq M \leq 6$.

M	$\beta_{\ell M}$	Minimizing N in (3.30)	β_{uM}	Maximizing N in (3.36)
3	2.666	3	4	∞
4	3.111	4	3.506	4
5	3.272	5	3.458	5
6	3.333	6	3.424	6
7	3.359	7	3.404	7

Table 3.4 Values of the coefficients $\beta_{\ell M}$ and β_{uM} in the bounds (3.30) and (3.36), respectively, on V_N.

To obtain an upper bound on V_N, we first observe that

$$L_i + L_{N-i} < L_N \quad , \qquad 0 < i < N. \tag{3.31}$$

A proof of (3.31) is unwardingly tedious and will be omitted, but its obviousness can be seen from the fact that it merely states that the CCRA will do better in processing two non-empty sets of transmitters in separate CRI's than it would do if the two sets first were merged and then the CCRA applied. In fact, it should be obvious from the tight bounds on L_N developed in the previous section that the right side of (3.31) will exceed the left by very close to unity as soon as both i and N - i are four or greater, and this can be made the basis

of a rigorous proof of (3.31). Next, we observe that (3.31) when $i = 1$ or $i = N - 1$ can be strengthened to

$$L_1 + L_{N-1} < L_N - 1, \tag{3.32}$$

as follows from the fact that $L_1 = 1$ so that the right side of 3.32 will exceed the left by approximately $\alpha - 2 \approx 0.9$ when N is four or greater — the validity of (3.32) for $N < 4$ can be checked from Table 3.1.

Using (3.31), (3.32) and the fact that $p_N(1) = Np_N(0) \geq 3p_N(0)$ for $N \geq 3$, one easily finds

$$\sum_{i=0}^{N} (1+L_i+L_{N-i})^2 p_N(i) < (L_N+1)^2 \quad , \quad N \geq 3. \tag{3.33}$$

Substituting (3.33) into (3.25) gives

$$V_N < \left[2 \sum_{i=0}^{N-1} V_i p_N(i)+2L_N+1\right] \Big/ (1-2^{-N+1}), \quad N \geq 3 \tag{3.34}$$

which is our desired simple upper bound on V_N. In the now familiar manner we can use (3.34) to verify that

$$V_N \leq \beta_{uM} N \quad , \quad N \geq M \tag{3.35}$$

holds for the choice

$$\beta_{uM} = \sup_{N \geq M} \left\{ \left[\sum_{i=0}^{M-1} \binom{N}{i} V_i + \alpha_M N - \frac{1}{2}\right] \Big/ \left[\sum_{i=0}^{M-1} \binom{N}{i} i\right] \right\} \tag{3.36}$$

for $M \geq 3$, where here α_M is any constant for which it is known that $L_N \leq \alpha_M N - 1$ for $N \geq M$. Taking $\alpha = 2.90$ for $M = 3$ and $\alpha = 2.8867$ for $M \geq 4$ (as were justified in the previous section), (3.36) results in the values of β_{uM} as given in Table 3.4.

The $M = 7$ cases in Table 3.4 provide the bounds

$$3.359N \leq V_N \leq 3.404N \quad , \quad N \geq 4. \tag{3.37}$$

The validity of (3.36) for $N < M$, i.e., for $N = 4$, 5 and 6 may be directly checked from Table 3.3. Although (3.37) is not quite so tight as our corresponding

bounds in (3.21) on L_N, it is tight enough to confirm our earlier suspicion that $\beta \approx 3.4$ and more than tight enough for our later computations. From (3.21), (3.37) and (3.4), we find the corresponding bounds on S_N to be

$$8.300N^2 - 2.403N+1 \leq S_N \leq 8.333N^2 - 2.369N+1, \quad N \geq 3. \qquad (3.38)$$

which, since the coefficient of N^2 is $\alpha_{\ell M}^2$ and α_{uM}^2 on the left and right sides, respectively, shows that the tightness of these bounds on S_N depends much more on the tightness of (3.21) than on that of (3.37).

3.5 Conditional Distribution of CRI Length

In this section, we consider the probability distributions, $P_{Y|X}(\ell|N)$, for the CRI length Y given that X = N transmitters collide in the first slot and the CCRA is used. The cases X = 0 and X = 1 are trivial and give

$$P_{Y|X}(1|0) = P_{Y|X}(1|1) = 1. \qquad (3.39)$$

For $N \geq 2$, however, every sufficiently large odd integer is a possible value of i. That Y must be odd follows from the facts that a binary rooted tree always has an odd number of nodes and that the slots in a CRI for the CCRA corresponds to the nodes in such a tree.

Writing $P(2m+1|N)$ for brevity in place of $P_{Y|X}(2m+1|N)$, we note first that

$$P(3|2) = 1/2, \qquad (3.40)$$

because with probability 1/2 the two transmitters colliding in slot 1 will flip different values and the CRI length is then 3. When these two transmitters flip the same value, then a blank slot together with another collision occurs and hence

$$P(2m+1|2) = \frac{1}{2} P(2(m-1)+1|2) \quad , \quad m \geq 2. \qquad (3.41)$$

Equation (3.41) is a first-order linear recursion whose solution for the initial condition (3.40) is

$$P(2m+1|2) = 2^{-m} \quad , \quad m \geq 1, \qquad (3.42)$$

which shows that $(Y-1)/2$ is geometrically distributed when $X = 2$.

When $X = 3$, the minimum value of Y is 5 and occurs when and only when the three transmitters do not all flip the same value after the initial collision and the two who flip the same value then flip different values after their subsequent collision. This reasoning gives

$$P(5|3) = \frac{3}{4}\frac{1}{2} = \frac{3}{8} . \tag{3.43}$$

In general, we see that

$$P(2m+1|3) = \frac{1}{4} P(2(m-1)+1|3) + \frac{3}{4} P(2(m-1)|2)$$

for $m \geq 3$. With the aid of (3.41) this can be rewritten

$$P(2m+1|3) - \frac{1}{4} P(2(m-1)+1|3) = 3(2^{-m-1}) \quad , \quad m \geq 3. \tag{3.44}$$

Equation (3.43) is another first-order linear recursion whose solution for the initial condition (3.43) is

$$P(2m+1|3) = 3(2^{-m}) - 6(4^{-m}) \quad , \quad m \geq 2. \tag{3.45}$$

Continuing this same approach, one could find $P(2m+1|N)$ for all N. However, the number of terms in the resulting expression doubles with each increase of N so that the calculation fast becomes unrewardingly tedious. The above distributions for $N \leq 3$, however, are sufficiently simple in form that we can and will make convenient use of them in what follows.

This completes our analysis of the traffic-independent properties of the CCRA, and we now turn our attention to the MCCRA.

3.6 Corresponding Properties for the Modified CCRA

We now briefly treat the traffic-independent properties of the modified CCRA (MCCRA) that was introduced in Section 2.4. Using the same notation as was used for the CCRA, we first note that (3.12) for the MCCRA becomes

$$L_N = 1 + 2 \sum_{i=0}^{N} L_i p_N(i) - p_N(0) \tag{3.46}$$

for $N \geq 2$, as follows again from Figure 3.2 and the fact that when none of the

transmitters in the initial collision flip 0 [which occurs with probability $p_N(0)$] then only $L_N - 1$ [rather than L_N] slots on the average are required to resolve the "collision" that was certain. Solving (3.46) for L_N gives the recursion

$$L_N = \left[1 + 2\sum_{i=0}^{N-1} L_i p_N(i) - p_N(0)\right] \Big/ (1-2^{-N+1}) \tag{3.47}$$

which holds for all $N \geq 2$. The initial conditions are again

$$L_0 = L_1 = 1. \tag{3.48}$$

In Table 3.5, we give the first few values of L_N as found from (3.47). Comparing Tables 3.1 and 3.5, we see that the main effect of the slots saved by eliminating collisions in the MCCRA is to reduce L_2 from 5 to 9/2, and that this 10% savings propagates only slightly diminished to L_N with $N > 2$. We also note from

N	L_N
0	1
1	1
2	9/2
3	7
4	9.643
5	12.314
6	14.985

Table 3.5 Expected CRI length for the MCCRA given N packets in the first slot

Table 3.5 that $L_N - L_{N-1} \approx 2.67$ for $N \geq 4$.

Using precisely the same techniques as in Section 3.3, one easily finds for the MCCRA that

$$\alpha_{\ell M} N - 1 \leq L_N \leq \alpha_{uM} N - 1 \quad , \quad N \geq M \tag{3.49}$$

where $\alpha_{\ell M}$ and α_{uM} are the infimum and supremum, respectively, for $N \geq M$ of the function

$$f_M(N) = \left[\sum_{i=0}^{M-1} \binom{N}{i} (L_i+1) - \frac{1}{2} \right] \Big/ \sum_{i=0}^{M-1} \binom{N}{i} i. \qquad (3.50)$$

For $M = 5$, these bounds become

$$2.6607N - 1 \leq L_N \leq 2.6651N - 1 \quad , \quad N \geq 4 \qquad (3.51)$$

where the fact that the bounds also hold for $N = M - 1 = 4$ can be checked directly from Table 3.5.

Equation (3.24), which gives S_N for the CCRA, is easily converted to apply to the MCCRA by noting that the only required change for the MCCRA is that $E^2(Y|X=N, X_0=0) = (L_0+L_N)^2$ rather than $(1+L_0+L_N)^2$ because of the eliminated slot, so that (3.24) is changed to

$$S_N = \sum_{i=0}^{N} [2V_i + (1+L_i+L_{N-i})^2] p_N(i) - (2L_N+3)2^{-N}. \qquad (3.52)$$

Beginning from (3.52), one can readily find the recursion for V_N analogous to (3.25), and then derive linear upper and lower bounds on V_N as was done in Section 3.4 for the CCRA. We shall, however, rest content with the bounds (3.51) on L_N, both because these are more important than those for V_N and also because our primary interest is in the CCRA rather than the MCCRA for the reason stated above at the end of Section 2.4.

4. RANDOM-ACCESS VIA COLLISION-RESOLUTION

4.1 The Obvious Random-Access Algorithm

We now consider the use of a collision-resolution algorithm as the key part of a random-access algorithm for the idealized situation described in Section 2.1. The principle is simple. To obtain a random-access algorithm from a collision-resolution algorithm, one needs only to specify the rule by which a transmitter with a new packet to send will determine the slot for its initial transmission. Thereafter, the transmitter uses the collision-resolution algorithm (if necessary) to determine when the packet should be retransmitted. One such first-time transmission rule is the obvious one: transmit a new packet in the first slot following the collision-resolution interval (CRI) in progress when it arrived. (Here we tacitly assume that no more than one new packet arrives at any transmitter while a CRI is in progress.) We shall refer to the random-access algorithm obtained in this way as the obvious random-access algorithm (ORAA) for the incorporated collision-resolution algorithm. That the ORAA may not be the ideal way to incorporate a collision-resolution algorithm is suggested by the fact that, after a very long CRI, a large number of packets will usually be transmitted in the next slot so that a collision there is virtually certain. Nonetheless, the ORAA is simple in concept and implementation, and is so natural that one would expect its analysis to yield useful insights.

For brevity, we shall write CORAA and MCORAA to denote the ORAA's incorporating the Capetanakis collision-resolution algorithm (CCRA) and the modified CCRA, respectively. We shall analyze the CORAA and the MCORAA in some detail before considering less obvious ways to incorporate collision-resolution algorithms in random-access algorithms.

4.2 Intuitive Stability Analysis

We assume that the random-access system is activated at time $t = 0$

with no backlog of traffic. The unit of time will be taken as one slot so that the i-th slot is the time interval $(i,i+1]$, $i = 0, 1,2,\ldots$. The new packet process is a counting process N_t giving the number of new packets that arrive at their transmitters in the interval $(0,t]$. Thus, $N_{t+T} - N_t$ is the number of new packets that arrive in the interval $(t,t+T]$. We assume that the new packet process is characterized by a constant λ, the new packet rate, such that, for all $t \geq 0$, $(N_{t+T}-N_t)/T$ will be close to λ with high probability when λT is large. The action of the random-access algorithm on the arrival process N_t generates another counting process W_t giving the number of packets that have arrived at their transmitters in the interval $(0,t]$ but have not been successfully transmitted in this interval. The random-access algorithm is said to be stable or unstable according as whether W_t remains bounded or grows without bound with high probability as t increases.

The stability properties of the CORAA can be ascertained from the following intuitive argument. From (3.21) and (3.37), we see that both the mean L_N and variance V_N of CRI length for the CCRA grow linearly with the number, N, of packets in the first slot. Thus, by Tchebycheff's inequality, the CRI length will be close to its mean $L_N \approx 2.89N$ with high probability when N is large. Now suppose some CRI begins in slot i and that W_i is large. By the first-time transmission rule for the CORAA, all W_i packets will be transmitted in the first slot of the CRI so that the CRI length Y will be close to $2.89\ W_i$ slots with high probability. But close to $\lambda Y = 2.89\lambda\ W_i$ new packets will with high probability arrive during the CRI so that

$$W_{i+Y} \approx 2.89\lambda\ W_i \tag{4.1}$$

with high probability. Thus, W_t will remain bounded with high probability, i.e., the CORAA will be stable, if

$$\lambda < \frac{1}{2.89} = .346 \text{ packets/slot.}$$

Conversely, provided only that the arrival process has "sufficient variability" so that W_i can in fact be large, (4.2) shows that the CORAA will be unstable for

$$\lambda > \frac{1}{2.89} = .346 \text{ packets/slot.}$$

The "sufficient variability" condition excludes deterministic arrival processes such as that with $N_i = i$ for $i = 1,2,3,\ldots$ which has $\lambda = 1$ and for which the CORAA is trivially stable. Other than for this restriction, the condition $\lambda < .346$ packet/slot is both necessary and sufficient for stability of the CORAA with very weak conditions on the arrival process N_t that we shall not attempt to make precise.

The same argument for the MCORAA, making use of (3.51) shows that

$$\lambda < \frac{1}{2.67} = .375 \text{ packets/slot}$$

is the corresponding sufficient condition for stability, and also a necessary condition when the arrival process has "sufficient variability." It is worth noting here that the upper limit of stability of .375 packets/slot for the MCORAA exceeds the $1/e = .368$ packets/slot "maximum throughput" of the unstable slotted-Aloha random-access algorithm.

In the following sections, we shall make the above stability arguments precise for the important case where the new packet process if Poisson. Along the way, we will determine the "delay-throughput characteristic" for the CORAA.

4.3 Dynamic Analysis of the CORAA

We now consider the case where the new packet process is a stationary Poisson point process, i.e., where $N_{t+T} - N_t$ is a Poisson random variable with mean λT for all positive T and all $t \geq 0$. Let Y_i and X_i denote the length and number of packets in the first slot, respectively, of the i-th CRI when the CORAA is applied, where Y_0 and X_0 correspond to the CRI beginning at $t = 0$. By assumption there are no packets awaiting transmission at $t = 0$ so that

$$X_0 = 0 \tag{4.2a}$$

$$Y_0 = 1. \tag{4.2b}$$

By the Poisson assumption, given that $Y_i = L$, X_{i+1} is a Poisson random variable with mean λL, i.e.,

$$P(X_{i+1}=N|Y_i=L) = \frac{(\lambda L)^N}{N!} e^{-\lambda L} \tag{4.3}$$

for $N = 0,1,2,\ldots$.

Because of (4.3) (which reflects the independent increments property of the Poisson new packet process), Y_{i+1} is independent of $Y_0,Y_1,\ldots Y_{i-1}$ when Y_i is given. Thus Y_0, Y_1, $Y_2,\ldots$ is a Markov chain, as is also the sequence X_0, X_1, $X_2,\ldots$. We now consider the "dynamic behavior" of these chains in the sense of the dependence of $E(X_i)$ and $E(Y_i)$ on i. We first note that the Poisson assumption implies

$$E(X_{i+1}|Y_i=L) = \lambda L, \tag{4.4}$$

which upon multiplication by $P(Y_i=L)$ and summing over L yields

$$E(X_{i+1}) = \lambda E(Y_i). \tag{4.5}$$

Equation (4.5) shows that finding the dependence of either $E(X_i)$ or $E(Y_i)$ on i determines the dependence of the other, so we will focus our attention on $E(X_i)$.

To illustrate our approach, we begin with the rather crude upper bound

$$L_N \leq 3N - 1 + 2\delta_{0N} - \delta_{1N}, \quad \text{all } N \geq 0 \tag{4.6}$$

for the CCRA, which follows from (3.18) with $M = 2$. But $E(Y_i|X_i=N) = L_N$ so that (4.6) implies

$$E(Y_i|X_i=N) \leq 3N - 1 + 2\delta_{0N} - \delta_{1N} \tag{4.7}$$

Multiplying by $P(X_i=N)$ and summing over N gives

$$E(Y_i) \leq 3\, E(X_i) - 1 + 2P(X_i=0) - P(X_i=1). \tag{4.8}$$

Now using (4.5) in (4.8), overbounding $P(X_i=0)$ by 1 and underbounding $P(X_i=1)$ by 0, we obtain the recursive inequality

$$E(X_{i+1}) - 3\lambda E(X_i) \leq \lambda. \tag{4.9}$$

When equality is taken in (4.9), we have a first-order linear recursion whose solution for the initial condition (4.2a) is an upper bound on $E(X_i)$, namely

$$E(X_i) \leq \frac{\lambda}{1-3\lambda}[1-(3\lambda)^i], \quad \text{all } i \geq 0. \tag{4.10}$$

showing that $E(X_i)$ approaches a finite limit as $i \to \infty$ provided $\lambda < \frac{1}{3}$. It is more interesting, however, to consider a "shift in the time origin" to allow

$$X_0 = N \tag{4.11}$$

to be an arbitrary initial condition. The solution of (4.9) then yields the bound

$$E(X_i) \leq N(3\lambda)^i + \frac{\lambda}{1-3\lambda}[1-(3\lambda)^i], \quad \text{all } i \geq 0. \tag{4.12}$$

Inequality (4.12) shows that when $\lambda < \frac{1}{3}$ and X_0 is very large (as when we take the time origin to be at a point where momentarily a large number of packets are awaiting their first transmission), $E(X_i)$ approaches its asympotic value of less than $\lambda/(1-3\lambda)$ exponentially fast in the CRI index i, and thus at least this fast in time t as the successive CRI's are decreasing in length on the average.

A similar argument beginning from the correspondingly crude lower bound for M = 2 in (3.18), namely

$$L_N \geq 2N - 1 + 2\delta_{0N} \tag{4.13}$$

would give, for the initial condition (4.11), the bound

$$E(X_i) \geq N(2\lambda)^i - \frac{\lambda}{1-2\lambda}[1-(2\lambda)^i], \quad \text{all } i \geq 0, \tag{4.14}$$

showing that $E(X_i) \to \infty$ as $i \to \infty$ when $\lambda > \frac{1}{2}$, and also showing for $\lambda < \frac{1}{2}$ that the approach of $E(X_i)$ to its asymptotic value is not faster than exponential in the CRI index i.

It should be clear that had bounds (4.6) and (4.13) been replaced by the correspondingly sharp bounds from (3.21), we would have found that

$$E(X_\infty) \overset{\Delta}{=} \lim_{i\to\infty} E(X_i) \tag{4.15}$$

is finite for

$$\lambda < \frac{1}{2.8867} = .3465 \qquad \text{(CORAA)} \tag{4.16}$$

but is infinite for

$$\lambda > \frac{1}{2.8810} = .3471. \qquad \text{(CORAA)} \tag{4.17}$$

Moreover, (4.5) implies that

$$E(Y_\infty) \overset{\Delta}{=} \lim_{i\to\infty} E(Y_i) = \frac{1}{\lambda} E(X_\infty) \tag{4.18}$$

so that (4.16) and (4.17) are also conditions for the finiteness and non-finiteness, respectively, of $E(Y_\infty)$.

Similar arguments based on (3.51) would have shown for the MCORRA that

$$\lambda < \frac{1}{2.6651} = .3752 \qquad \text{(MCORAA)} \tag{4.19}$$

and

$$\lambda > \frac{1}{2.6607} = .3758 \qquad \text{(MCORAA)} \tag{4.20}$$

imply the finiteness and non-finiteness, respectively, of $E(Y_\infty)$.

We will shortly see that conditions (4.16) and (4.17) in fact imply the stability and instability, respectively, of the CORAA corroborating the intuitive analysis of Section 4.2. Conditions (4.19) and (4.20) similarly imply the stability and instability, respectively, of the MCORAA.

4.4 Stability Analysis of the CORAA

We have just seen that the Markov chain $X_0, X_1, X_2, \ldots$ [giving the number of packets in the first slots of the CRI's when the new packet traffic is Poisson and the CORAA is used] has $E(Y_\infty) < \infty$ when $\lambda < .3465$. We also note from (4.3) that, regardless of the value of X_i, X_{i+1} has nonzero probability of being any nonnegative integer. These two facts imply that for $\lambda < .3465$ the chain has

steady-state probabilities

$$\pi_N = P(X_\infty = N) \stackrel{\Delta}{=} \lim_{i\to\infty} P(X_i = N) \qquad N = 0,1,2,\ldots \tag{4.21}$$

and is ergodic in the sense that if n_N is the number of CRI's with N packets among the first n CRI's, then

$$\lim_{n\to\infty} \frac{n_N}{n} = \pi_N \quad \text{(a.s.)}. \tag{4.22}$$

Similarly, $\lambda < .3465$ implies that the Markov chain $Y_0, Y_1 Y_2, \ldots$ has steady-state probabilities

$$P(Y_\infty = L) \stackrel{\Delta}{=} \lim_{i\to\infty} P(Y_i = L) \qquad L = 1,3,5,\ldots \tag{4.23}$$

such that

$$\lim_{n\to\infty} \frac{n_L'}{n} = P(Y_\infty = L) \tag{4.24}$$

where n_L' is the number of CRI's of length L among the first n CRI's.

Let the random variable Y_a denote the length of the CRI in progress when a "randomly-chosen packet" arrives at its transmitter. Because the new packet arrival process is stationary, $P(Y_a = L)$ will equal the fraction of the time axis occupied by CRI's of length L, i.e.,

$$P(Y_a = L) = \lim_{n\to\infty} \frac{L n_L'}{\sum_{i=1}^{\infty} i\, n_i'}, \quad \text{(a.s.)}. \tag{4.25}$$

Dividing by n in the numerator and denominator on the right of (4.25) and making use of (4.24) gives

$$P(Y_a = L) = \frac{L P(Y_\infty = L)}{E(Y_\infty)}. \tag{4.26}$$

Multiplying in (4.26) by L and summing over L gives

$$E(Y_a) = \frac{E(Y_\infty^2)}{E(Y_\infty)}, \tag{4.27}$$

where

$$E(Y_\infty^2) \triangleq \lim_{i\to\infty} E(Y_i^2) \qquad \text{(a.s.)} \tag{4.28}$$

as follows again from ergodicity.

Now let the random variable Y_d denote the length of the CRI in which the same randomly chosen packet departs from the system in the sense of being successfully transmitted, and let X_d be the total number of packets in this CRI. From (4.4) and the fact that in the CORAA a packet departs in the CRI immediately following that in which it arrives, we have

$$E(X_d|Y_a=L) = \lambda L. \tag{4.29}$$

Multiplying by $P(Y_a=L)$ and summing gives

$$E(X_d) = \lambda E(Y_a). \tag{4.30}$$

Next, we note that

$$E(Y_d|X_d=N) = L_N \leq 2.8867N + \delta_{0N} - 1.8867\delta_{1N} \tag{4.31}$$

is a simple but rather tight upper bound which follows from (3.21) and a check of the cases N = 2 and N = 3. Multiplying by $P(X_d=N)$ in (4.31) and summing over N gives

$$E(Y_d) \leq 2.8867E(X_d) + P(X_d=0) - 1.8867P(X_d=1). \tag{4.32}$$

But $P(X_d=1|Y_a=L) = \lambda Le^{-\lambda L} = \lambda LP(X_d=0|Y_a=L) \geq \lambda P(X_d=0|Y_a=L)$ for all L which implies

$$P(X_d=1) \geq \lambda P(X_d=0). \tag{4.33}$$

Using (4.33) in (4.32) gives

$$E(Y_d) \leq 2.8867E(X_d) + (1-1.8867\lambda)\, P(X_d=0). \tag{4.34}$$

But $P(X_d=0|Y_a=L) = e^{-\lambda L} \leq e^{-\lambda}$ for all L so that

$$P(X_d=0) \leq e^{-\lambda}. \tag{4.35}$$

Substituting (4.30) and (4.35) in (4.34) now gives

$$E(Y_d) \leq 2.8867\lambda E(Y_a) + (1-1.8867\lambda)e^{-\lambda} \tag{4.36}$$

provided $\lambda < .53$, which includes all λ in the range of interest as will soon be seen. Inequality (4.36) is our desired tight upper bound on $E(Y_d)$ in terms of $E(Y_a)$.

Similarly, starting from

$$L_N \geq 2.8810N - 1 + 2\delta_{0N} - 0.8810\delta_{1N}, \tag{4.37}$$

which follows from (3.21), we note that the same argument that led to (4.32) now gives

$$E(Y_d) \geq 2.8810E(X_d) - 1 + 2P(X_d=0) - 0.8810P(X_d=1).$$

Overbounding $P(X_d=1)$ by $1 - P(X_d=0)$ gives

$$E(Y_d) \geq 2.881E(X_d) - 1.881 + 2.881P(X_d=0). \tag{4.38}$$

But $P(X_d=0|Y_a=L) = e^{-\lambda L}$; hence multiplying by $P(Y_a=L)$, summing over L, and using Jensen's inequality [8] gives

$$P(X_d=0) \geq e^{-\lambda E(Y_a)}. \tag{4.39}$$

Substituting (4.39) in (4.38) and making use of (4.30) yields

$$E(Y_d) \geq 2.881\lambda E(Y_a) - 1.881 + 2.881e^{-\lambda E(Y_a)}, \tag{4.40}$$

which is our desired lower bound on $E(Y_d)$ in terms of $E(Y_a)$.

We now introduce the crucial random variable in a random-access system, namely the delay D experienced by a randomly-chosen packet, i.e., the time difference between its arrival at the transmitter and the onset of its successful transmission (so that $D = 0$ when the packet is successfully transmitted beginning at the same moment that it arrives at the transmitter). Now making precise the notion of stability introduced intuitively in Section 4.2, we say that the random-access system is stable just when $E(D) < \infty$.

For the CORAA, we first note that

$$\frac{1}{2} E(Y_a) + \frac{1}{2} E(Y_d-1) \leq E(D) \leq \frac{1}{2} E(Y_a) + E(Y_d-1) \tag{4.41}$$

as follows from the facts (i) that on the average the randomly-chosen packet

arrives at the midpoint of the CRI in progress, and (ii) that at the latest its successful transmission begins in the last slot of its departure CRI, but on the average somewhat beyond the midpoint of the slot starting times as follows from the discussion in Section 2.3. Substituting (4.40) in (4.41) gives the lower bound

$$E(D) \geq \frac{1}{2} + 1.4405\lambda \;\; E(Y_a) - 1.4405(1-e^{-\lambda E(Y_a)}). \qquad (4.42)$$

Similarly, using (4.36) in (4.41) gives the upper bound

$$E(D) \leq \frac{1}{2} + 2.8867\lambda \;\; E(Y_a) + (1-1.8867\lambda)e^{-\lambda}-1. \qquad (4.43)$$

From (4.42) and (4.43), it follows that the CORAA is stable if and only if $E(Y_a) < \infty$. We are thus motivated to explore (4.27) more closely.

We first note that $E(Y_\infty^2) \geq E^2(Y_\infty)$ implies by virtue of (4.27) that

$$E(Y_a) \geq E(Y_\infty), \qquad (4.44)$$

which in turn, because of (4.17), implies that <u>the CORAA is unstable for $\lambda > .3471$</u>.

We now proceed to obtain a rather tight upper bound on $E(Y_a)$. We begin with the bound

$$S_N \leq 8.333N^2 - 2.369N + 1 - 5.964\delta_{1N} + 3.406\delta_{2N} \qquad (4.45)$$

which follows from (3.38) and the facts that $S_0 = S_1 = 1$ and $S_2 = 33$. By virtue of the steady-state probabilities in the corresponding Markov chains, we have

$$E(Y_\infty^2) = \sum_{N=0}^{\infty} E(Y^2|X=N)\; P(X_\infty=N)$$

$$= \sum_{N=0}^{\infty} S_N\pi_N. \qquad (4.46)$$

Thus, multiplying by π_N in (4.43) and summing gives

$$E(Y_\infty^2) \leq 8.333E(X_\infty^2) - 2.369E(X_\infty) + 1 - 5.964\pi_1 + 3.406\pi_2 \qquad (4.47)$$

which is a very tight bound as follows from the tightness of (4.45). But from (4.3) and the fact that the mean and variance of a Poisson random variable

coincide, we have

$$E(X_{i+1}^2 | Y_i = L) = \lambda L + (\lambda L)^2. \tag{4.48}$$

Multiplying by the steady state probabilities $P(Y_\infty = L)$ and summing over L now gives

$$E(X_\infty^2) = \lambda E(Y_\infty) + \lambda^2 E(Y_\infty^2),$$

which, because of (4.18), can be written as

$$E(X_\infty^2) = E(X_\infty) + \lambda^2 E(Y_\infty^2). \tag{4.49}$$

Substituting (4.49) in (4.47) and rearranging gives

$$(1-8.333\lambda^2)\ E(Y_\infty^2) \le 5.964 E(X_\infty) + 1 - 5.964\pi_1 + 3.406\pi_2.$$

Dividing now by $E(Y_\infty) = E(X_\infty)/\lambda$ and using (4.27) yields

$$(1-8.333\lambda^2)\ E(Y_a) \le 5.964\lambda + \frac{1-5.964\pi_1+3.406\pi_2}{E(Y_\infty)}\ . \tag{4.50}$$

We now turn our attention to the second term on the right of (4.50).

First, we observe that

$$\begin{aligned} E(Y_\infty) &= \sum_{i=0}^{\infty} L_N \pi_N \\ &= 1 + \sum_{i=0}^{\infty} (L_N - 1)\ \pi_i \ge 1 + 4\pi_2 \end{aligned} \tag{4.51}$$

because $L_N \ge 1$ for all N and $L_2 = 5$. But (4.51) thus implies

$$\frac{1+3.406\pi_2}{E(Y_\infty)} \le 1, \tag{4.52}$$

which we shall shortly use in (4.50).

To handle the term involving π_1 in (4.50) requires more care. We begin somewhat indirectly by noting that multiplying by $\pi_N = P(X_\infty = N)$ in the upper bound (4.31) on $L_N = E(Y_\infty | X_\infty = N)$ and then summing over N gives

$$E(Y_\infty) \le 2.8867 E(X_\infty) + \pi_0 - 1.8867\pi_1. \tag{4.53}$$

Overbounding π_0 by $1 - \pi_1$ and using (4.18) gives

$$E(Y_\infty) \le \frac{1-2.8867\pi_1}{1-2.8867\lambda} \tag{4.54}$$

provided $\lambda < .3464$. The rather tight bound (4.54) is of some interest in itself. We first use (4.54) only to note that the right side is less than 1 if $\pi_1 > \lambda$; but $E(Y_\infty) \ge 1$ so that by contradiction we conclude that

$$\pi_1 \le \lambda \tag{4.55}$$

for $\lambda < .3464$. In Section 4.5 we will show that

$$\pi_1 \ge \lambda(1-\lambda), \qquad \lambda \le .22 \tag{4.56}$$

which indicates the tightness of (4.55) and (4.56).

Next, we note that (4.54) implies

$$\frac{\pi_1}{E(Y_\infty)} \ge \frac{\pi_1}{1-\alpha\pi_1}(1-\alpha\lambda) \tag{4.57}$$

where for convenience we have written

$$\alpha = 2.8867. \tag{4.58}$$

The right side of (4.57) increases with π_1. Thus, using (4.56) in (4.57) gives

$$\frac{\pi_1}{E(Y_\infty)} \ge \frac{\lambda(1-\lambda)(1-\alpha\lambda)}{1-\alpha\lambda(1-\lambda)}, \qquad \lambda \le .22 \tag{4.59}$$

Now using (4.59) for $\lambda \le .22$ and the trivial bound $\pi_1/E(Y_\infty) \ge 0$ for $\lambda \ge .22$ together with (4.52) in (4.50), we obtain

$$E(Y_a) \le \begin{cases} \dfrac{5.964\lambda + 1 - 5.964\lambda(1-\lambda)(1-2.8867\lambda)/[1-2.8867\lambda(1-\lambda)]}{1 - 8.333\lambda^2}, & \lambda \le .22 \\[2ex] \dfrac{5.964\lambda + 1}{1 - 8.333\lambda^2}, & .22 < \lambda < .3464 \end{cases} \tag{4.60}$$

which is our desired rather tight upper bound on $E(Y_a)$. Because $8.333\lambda^2 = (2.8867\lambda)^2$, (4.60) verifies that <u>the CORAA is stable for $\lambda < .3464$</u>, as was anticipated in (4.16).

An entirely similar argument beginning from

$$S_N \ge 8.300N^2 - 2.403N + 1 - 5.897\delta_{1N} + 3.606\delta_{2N} \tag{4.61}$$

[which follows from (3.38)], rather than from (4.45), leads to

$$(1-8.300\lambda^2)E(Y_a) \geq 5.897\lambda + \frac{1 - 5.897\pi_1 + 3.606\pi_2}{E(Y_\infty)} \tag{4.62}$$

rather than to (4.50). The tightness of the bounds (4.50) and (4.62) is evident. Underbounding π_2 by 0 and using (4.55) in (4.62) gives

$$(1-8.300\lambda^2)E(Y_a) \geq 5.897\lambda + \frac{1 - 5.897\lambda}{E(Y_\infty)} . \tag{4.63}$$

When $\lambda > .1696$ so that $1 - 5.897\lambda < 0$, we can use the trivial bound $E(Y_\infty) \geq 1$ to see that the right side of (4.63) is underbounded by 1. When $\lambda \leq 1.696$ we can use (4.54) and (4.56) to show that the right side of (4.63) is underbounded by

$$5.897\lambda + (1-5.897\lambda)(1-2.8867\lambda)/[1-2.8867\lambda(1-\lambda)].$$

Combining these two bounds, we have

$$E(Y_a) \geq \begin{cases} \dfrac{5.897\lambda + (1-5.897\lambda)(1-2.8867\lambda)/[1-2.8867\lambda(1-\lambda)]}{1 - 8.300\lambda^2}, & \lambda \leq .1696 \\[2ex] \dfrac{1}{1 - 8.300\lambda^2}, & .1696 < \lambda < .3464 \end{cases} \tag{4.64}$$

Inequality (4.64) is our desired lower bound on $E(Y_a)$.

Table 4.1 gives a short tabulation of the upper and lower bounds (4.60) and (4.64), respectively, on $E(Y_a)$. The relative tightness of these bounds is perhaps more visible in Figure 4.1. A close review of the bounding arguments suggests that the upper bound (4.60) is a better approximation to $E(Y_a)$ than is the lower bound (4.64).

Using (4.60) in (4.43), we obtain the upper bound on the expected delay, E(D), of a randomly-chosen packet as tabulated in Table 4.1. We could use (4.64) together with (4.42) to get a lower bound on E(D). However, we recall that the tightness of the upper bound (4.60) suggests instead using it together with (4.42) to obtain the "approximate lower bound" tabulated in Table 4.1.

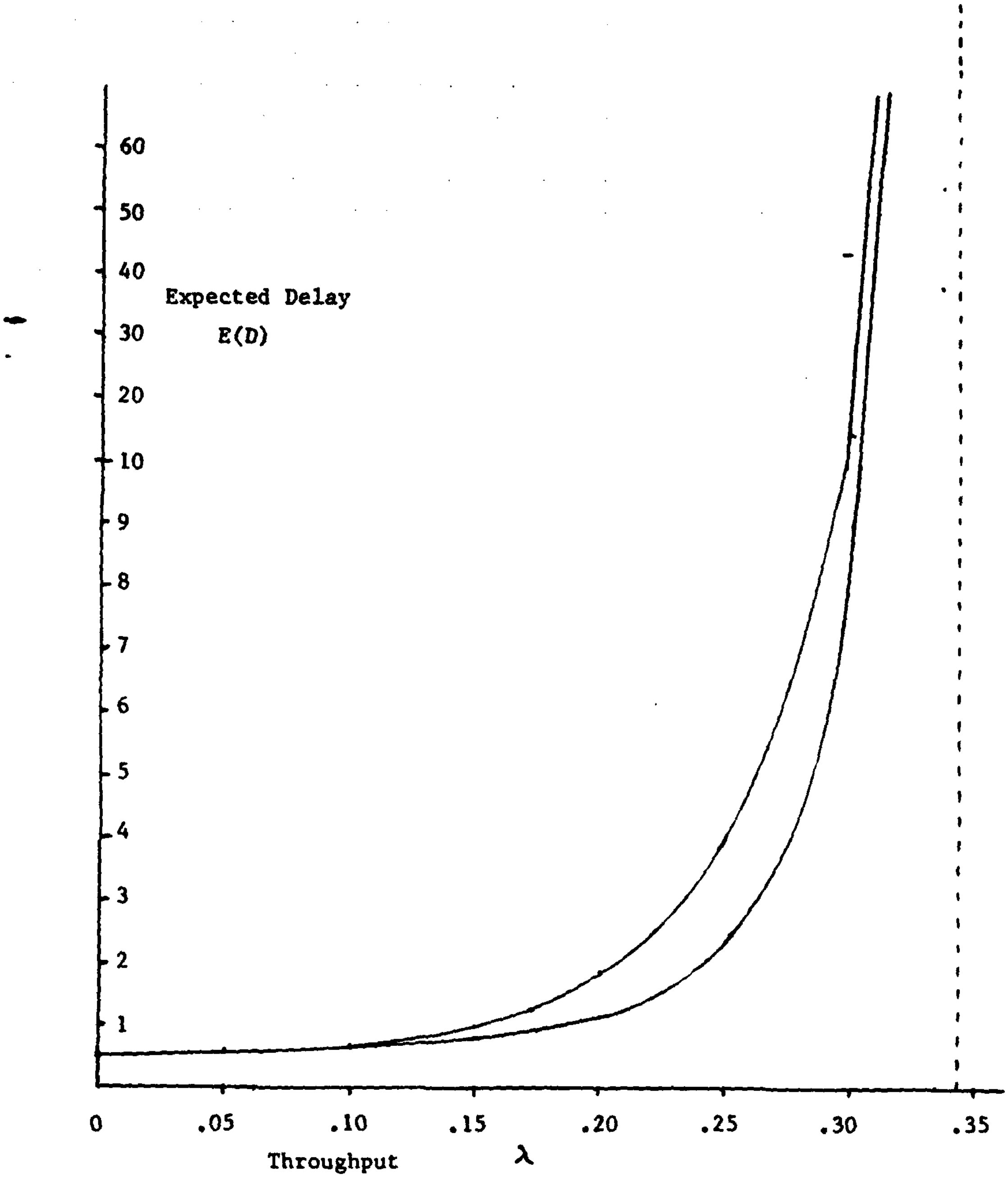


Fig. 4.1: The Upper Bound (4.43) on Expected Delay of a Randomly-Chosen Packet for the CCRA, and an Approximate Lower Bound. (Note change of scale at E(D) = 10.)

λ	$E(Y_a)$ upper bound (4.60)	$E(Y_a)$ lower bound (4.64)	E(D) upper bound	E(D) approx. lower bound
0	1	1	1/2	1/2
.05	1.039	1.015	.531	.521
.10	1.179	1.073	.664	.599
.15	1.492	1.216	1.009	.780
.1696	1.696	1.315	1.252	.902
.20	2.165	1.497	1.842	1.200
.25	3.783	2.078	4.03	2.37
.30	11.16	3.952	14.6	9.01
1/3	40.32	12.86	58.2	38.1
.34	82.49	24.68	121.5	80.2
.345	374.5	82.7	559.5	371.9

Table 4.1: Upper and lower bound on the expected length, $E(Y_a)$, of the CRI in which a randomly-chosen packet arrives versus the throughput λ, and on the expected delay for a randomly-chosen packet, for the CORAA.

Our purpose is to illustrate that the bounds (4.42) and (4.43) do not significantly differ since their common term $\frac{1}{2} E(Y_a)$ is the dominant one. In Figure 4.1, we have plotted this approximate lower bound on E(D) together with the strict upper bound. As λ is the throughput of the system, the plot of E(D) versus λ is the delay-throughput characteristic of the MCORAA. Figure 4.1 gives a strict upper bound on this characteristic together with an approximate lower bound which indicates that this upper bound is quite tight.

4.5 Steady-State Probabilities for the CORAA

We have already in (4.21) introduced the steady-state probabilities

$$\pi_N = P(X_\infty = N) \quad , \quad N = 0,1,2,\ldots$$

for the CORAA. The equilibrium equations satisfied by these steady-state

probabilities are

$$\pi_N = \sum_{n=0}^{\infty} P_{Nn}\pi_n \ , \ N = 0,1,2,\ldots \tag{4.65}$$

where P_{Nn} is the transition probability

$$\begin{aligned} P_{Nn} &= P(X_{i+1}=N|X_i=n) \\ &= \sum_{L=1}^{\infty} P(X_{i+1}=N|Y_i=L)\ P(Y_i=L|X_i=n) \\ &= \sum_{L=1}^{\infty} \frac{(\lambda L)^N}{N!} e^{-\lambda L} P(Y_i=L|X_i=n) \end{aligned} \tag{4.66}$$

where we have made use of (4.3). In general, these transition probabilities are difficult to calculate. However, because of (3.39), (4.66) for n = 0 and 1 becomes

$$P_{N0} = P_{N1} = \frac{\lambda^N}{N!} e^{-\lambda}, \quad N = 0,1,2,\ldots \ . \tag{4.67}$$

In Section 3.5, we calculated $P(Y_i=L|X_i=2)$. We now use this distribution, given by (3.42), in (4.66) to obtain by summing the resultant series

$$\begin{aligned} P_{02} &= \sum_{m=1}^{\infty} e^{-\lambda(2m+1)} 2^{-m} \\ &= e^{-3\lambda}/(2-e^{-2\lambda}) \triangleq A(\lambda), \end{aligned} \tag{4.68}$$

$$\begin{aligned} P_{12} &= \sum_{m=1}^{\infty} \lambda(2m+1) e^{-\lambda(2m+1)} 2^{-m} \\ &= A(\lambda)\lambda(6-e^{-2\lambda})/(2-e^{-2\lambda}) \\ &\triangleq B(\lambda), \end{aligned} \tag{4.69}$$

and

$$\begin{aligned} P_{22} &= \sum_{m=1}^{\infty} \frac{1}{2} \lambda^2 (2m+1)^2 e^{-\lambda(2m+1)} 2^{-m} \\ &= \frac{1}{2} \lambda^2 A(\lambda) [1+32/(2-e^{-2\lambda})^2] \\ &\triangleq C(\lambda). \end{aligned} \tag{4.70}$$

As we shall soon see, these few explicit transition probabilities are quite

enough to establish the bound (4.56), which is the main objective of this section.

Making use of (4.67)-(4.70), we can write (4.65) for N = 1, 2, and 3 as

$$\pi_0 = e^{-\lambda}(\pi_0+\pi_1) + A(\lambda)\pi_2 + \dots \tag{4.71a}$$

$$\pi_1 = \lambda e^{-\lambda}(\pi_0+\pi_1) + B(\lambda)\pi_2 + \dots \tag{4.716b}$$

$$\pi_2 = \frac{1}{2}\lambda^2 e^{-\lambda}(\pi_0+\pi_1) + C(\lambda)\pi_2 + \dots \tag{4.71c}$$

where the terms not shown explicitly are of course nonnegative. To proceed further, we need to make use of certain bounds on $E(Y_\infty)$. First, we note that

$$E(Y_\infty) = \sum_{N=0}^{\infty} L_N \pi_N$$

$$\geq \pi_0 + \pi_1 + 5\pi_2 + \frac{23}{3}(1-\pi_0-\pi_1-\pi_2)$$

where we have made use of Table 3.1 and the fact that L_N increases with N. Thus, we have

$$3E(Y_\infty) \geq 23 - 20(\pi_0+\pi_1) - 8\pi_2. \tag{4.72}$$

Discarding the nonnegative higher order terms on the right in (4.71) gives

$$\pi_0 \geq e^{-\lambda}(\pi_0+\pi_1) + A(\lambda)\pi_2 \tag{4.73a}$$

$$\pi_1 \geq \lambda e^{-\lambda}(\pi_0+\pi_1) + B(\lambda)\pi_2 \tag{4.73b}$$

$$\pi_2 \geq \frac{1}{2}\lambda^2 e^{-\lambda}(\pi_0+\pi_1) + C(\lambda)\pi_2. \tag{4.73c}$$

Summing (4.73a) and (4.73b), then rearranging, gives

$$\pi_2 \leq F(\lambda)(\pi_0+\pi_1) \tag{4.74}$$

where

$$F(\lambda) \triangleq \frac{1-(\lambda+1)e^{-\lambda}}{A(\lambda) + B(\lambda)}. \tag{4.75}$$

Using (4.74) in (4.72) gives

$$3E(Y_\infty) \geq 23 - [20+8F(\lambda)](\pi_0+\pi_1). \tag{4.76}$$

But (4.73c) is equivalent to

$$\pi_2 \geq G(\lambda)(\pi_0+\pi_1) \tag{4.77}$$

where

$$G(\lambda) \triangleq \frac{1}{2}\lambda^2 e^{-\lambda}/[1-C(\lambda)]. \tag{4.78}$$

Using (4.77) in (4.73b) gives

$$\pi_1 \geq [\lambda e^{-\lambda}+B(\lambda)G(\lambda)](\pi_0+\pi_1). \tag{4.79}$$

Now using (4.79) in (4.76) gives

$$3E(Y_\infty) \geq 23 - \pi_1[20+8F(\lambda)]/[\lambda e^{-\lambda}+B(\lambda)G(\lambda)]. \tag{4.80}$$

Inserting the upper bound of (4.54) on $E(Y_\infty)$ into (4.80) and rearranging, we obtain

$$\pi_1 \geq (20-23a\lambda)/H(\lambda), \quad \text{if} \quad H(\lambda) > 0 \tag{4.81}$$

where

$$H(\lambda) \triangleq \frac{(1-a\lambda)[20+8F(\lambda)]}{\lambda e^{-\lambda} + B(\lambda)G(\lambda)} - 3a \tag{4.82}$$

and

$$a = 2.8867.$$

Inequality (4.81) is our desired lower bound on π_1 and is tabulated in Table 4.2. We see that

$$\pi_1 \geq \lambda(1-\lambda) \qquad \text{for} \qquad \lambda \leq .2209 \tag{4.83}$$

which is our desired justification of (4.56).

Proceeding in a similar way, we can derive lower bounds on π_0 and π_2. Using (4.79) in the bound (4.54), we obtain

$$3E(Y_\infty) \leq 3\{1-a[\lambda e^{-\lambda}+B(\lambda)G(\lambda)](\pi_0+\pi_1)\}/(1-a\lambda). \tag{4.84}$$

Combining (4.76) and (4.84), we obtain after some rearrangement

$$\pi_0 + \pi_1 \geq J(\lambda) \tag{4.85}$$

where

$$J(\lambda) \triangleq \frac{20 - 23a\lambda}{(1-a\lambda)[20+8F(\lambda)] - 3a[\lambda e^{-\lambda}+B(\lambda)G(\lambda)]} \tag{4.86}$$

provided the denominator of (4.86) is positive. Inserting (4.85) and (4.77) on the right in (4.73a) now gives

$$\pi_0 \geq [e^{-\lambda}+A(\lambda)G(\lambda)]J(\lambda) \ , \tag{4.87}$$

which is our desired lower bound on π_0. Similarly, using

λ	$\lambda(1-\lambda)$	Lower Bounds (4.87) π_0	Lower Bounds (4.81) π_1	Lower Bounds (4.88) π_2	Upper Bounds (4.106) π_0	Upper Bounds (4.95) π_1	Upper Bounds (4.104) π_2
0	0	1	0	0	1	0	0
.05	.0475	.9506	.04770	.00122	.9510	.04775	.00125
.10	.0900	.9011	.0911	.00485	.9034	.0916	.00516
.15	.1275	.8488	.1299	.0108	.8564	.1319	.0122
.20	.1600	.7872	.1620	.0185	.8094	.1686	.0230
.2209	.1721	.7545	.1721	.0219	.7896	.1828	.0288
.25	.1875	.6909	.1791	.0262	.7620	.2015	.0382
.28	.2016	.5462	.1592	.0263	.7332	.2193	.0497
.30	.2100	.0870	.0272	.0048	.7139	.2303	.0584
.3464	-	-	-	-	.6689	.2533	.0820

Table 4.2: Upper and Lower Bounds on the Steady-State Probabilities $\pi_N = P(X_\infty=N)$ for the CORAA versus the throughput λ.

(4.86) in (4.77) gives the bound

$$\pi_2 \geq G(\lambda)J(\lambda). \tag{4.88}$$

A short tabulation of the bounds (4.87) and (4.88) is included in Table 4.2.

Next, we turn our attention to overbounding the steady-state probabilities. Beginning with π_1, we first note that because

$$xe^{-x} \leq e^{-1} \quad , \text{ all } x, \tag{4.89}$$

it follows from (4.3) that

$$P(X_{i+1}=1|Y_i=L) \leq e^{-1}, \quad \text{all } L, \tag{4.90}$$

and hence from (4.66) that

$$P_{1n} \leq e^{-1} \ , \quad \text{all } n. \tag{4.91}$$

Using (4.91) to overbound the terms for $n \geq 3$ on the right in (4.65) gives

$$\pi_1 \leq \lambda e^{-\lambda}(\pi_0+\pi_1) + B(\lambda)\pi_2 + e^{-1}(1-\pi_0-\pi_1-\pi_2). \tag{4.92}$$

A simple check shows that $B(\lambda) \leq e^{-1}$ for $0 \leq \lambda \leq 1$ so that (4.77) can be used on the right in (4.92) to give

$$\pi_1 \leq e^{-1} - [e^{-1}-\lambda e^{-\lambda}+e^{-1}G(\lambda) - B(\lambda)G(\lambda)](\pi_0+\pi_1). \tag{4.93}$$

To proceed further, we need to overbound $\pi_0 + \pi_1$ in terms of π_1. Adding π_1 to both sides of (4.73a) and using (4.77), we obtain

$$\pi_0 + \pi_1 \geq \pi_1/[1-e^{-\lambda}-A(\lambda)G(\lambda)] \tag{4.94}$$

where we have made use of the fact that $A(\lambda) < 1 - e^{-\lambda}$ for $0 < \lambda \leq 1$. Now using (4.94) in (4.93), we obtain our desired upper bound

$$\pi_1 \leq J(\lambda) \tag{4.95}$$

where

$$J(\lambda) \triangleq e^{-1}/\{1+[e^{-1}-\lambda e^{-\lambda}+e^{-1}G(\lambda)-B(\lambda)G(\lambda)]/[1-e^{-\lambda}-A(\lambda)G(\lambda)]\}, \tag{4.96}$$

provided the expression in wavy brackets is positive. The bound (4.95) is tabulated in Table 4.2.

To obtain upper bounds on π_0 and π_1, we begin by first noting from (4.3) that

$$P(X_i=0|Y_i=L) = e^{-\lambda L} \tag{4.97}$$

But, as we saw in Section 3, $X_i \geq 3$ implies $Y_i \geq 5$ because the CRI must contain at least 2 collisions. Thus, (4.94) and (4.97) imply

$$P_{ON} \leq e^{-5\lambda}, \quad N \geq 3. \tag{4.98}$$

Using (4.98) on the right in (4.65) gives

$$\pi_0 \leq e^{-\lambda}(\pi_0+\pi_1) + A(\lambda)\pi_2 + e^{-5\lambda}(1-\pi_0-\pi_1-\pi_2). \tag{4.99}$$

Summing (4.92) and (4.99) gives

$$\pi_0 + \pi_1 \leq e^{-1} + e^{-5\lambda} + [(\lambda+1)e^{-\lambda}-e^{-1}-e^{-5\lambda}](\pi_0+\pi_1) - [e^{-1}+e^{-5\lambda}-A(\lambda)-B(\lambda)]\pi_2. \tag{4.100}$$

A simple check shows that $A(\lambda) + B(\lambda) \leq e^{-1} + e^{-5\lambda}$ holds for $0 \leq \lambda \leq .1779$

and $.3179 \leq \lambda \leq 1$; thus, we can in this range use (4.77) on the right in (4.100) to obtain

$$\pi_0 + \pi_1 \leq K_1(\lambda) \quad \text{for} \quad 0 \leq \lambda \leq .1779 \quad \text{and} \quad .3179 \leq \lambda \leq 1, \tag{4.101}$$

where

$$K_1(\lambda) = (e^{-1}+e^{-5\lambda})/[1+e^{-1}+e^{-5\lambda}-(\lambda+1)e^{-\lambda} + (e^{-1}+e^{-5\lambda})G(\lambda)-A(\lambda)G(\lambda)-B(\lambda)G(\lambda)]. \tag{4.102}$$

Similarly, for $.1779 < \lambda < .3179$ we can use (4.74) on the right in (4.100) to obtain

$$\pi_0 + \pi_1 \leq K_2(\lambda) \quad \text{for} \quad .1779 < \lambda < .3179 \tag{4.103}$$

where $K_2(\lambda)$ is equal to the right side of (4.102) with $G(\lambda)$ replaced by $F(\lambda)$. It now follows from (4.74) that

$$\pi_2 \leq \begin{cases} F(\lambda)K_1(\lambda) & \text{for} \quad 0 \leq \lambda \leq .1779 \quad \text{and} \quad .3179 \leq \lambda \leq 1 \\ F(\lambda)K_2(\lambda) & \text{for} \quad .1779 < \lambda < .3179. \end{cases} \tag{4.104}$$

This bound is also tabulated in Table 4.2. Finally, we note that using (4.74) in (4.99) gives

$$\pi_0 \leq e^{-5\lambda} + (e^{-\lambda}-e^{-5\lambda})(\pi_0+\pi_1) + [A(\lambda)-e^{-5\lambda}](\pi_0+\pi_1) \tag{4.105}$$

where we have used the fact that $A(\lambda) \geq e^{-5\lambda}$ for $0 \leq \lambda \leq 1$. Using (4.105) with (4.101) and (4.103) then gives

$$\pi_0 \leq e^{-5\lambda} + [e^{-\lambda}-e^{-5\lambda}+F(\lambda)A(\lambda)-F(\lambda)e^{-5\lambda}]K_i(\lambda) \tag{4.106}$$

where

$$i = \begin{cases} 1 & \text{for} \quad 0 \leq \lambda \leq .1779 \quad \text{and} \quad .3179 \leq \lambda \leq 1 \\ 2 & \text{for} \quad .1779 < \lambda < 1. \end{cases} \tag{4.107}$$

The bound (4.106) is also tabulated in Table 4.2.

From Table 4.2, we see that our upper and lower bounds on π_0, π_1 and π_2 are so tight for $\lambda \leq .15$ as to be virtually equalities, and still reasonably tight for $.15 < \lambda \leq .25$. The lower bounds begin to degrade rapidly, however, for $\lambda \geq .28$. The chief reason for this is that the upper bound (4.54) on

$E(Y_\infty)$ that was used to obtain the lower bounds on π_0, π_1 and π_2 becomes very loose in the region $\lambda \geq .28$. To improve our lower bounds, we should make use of (4.50) to obtain an upper bound on $E(Y_\infty)$ that will involve both π_1 and π_2, then use this bound in place of (4.54) in the argument that led to the lower bounds on π_0, π_1, and π_2. To obtain even tighter lower bounds, we could begin from an upper bound on $E(Y_\infty)$ in terms of π_1, π_2 and π_3. Etc.

The upper bounds on π_0, π_1 and π_2 above, however, did not utilize any bounds on $E(Y_\infty)$ in their derivations. These upper bounds should be virtual equalities for $0 \leq \lambda \leq .30$ as the inequalities introduced to obtain them are all extremely tight for this range of λ. To tighten these bounds further, we would need to include additional explicit transition probabilities in (4.73).

It should be clear from this section that any finite number of the steady-state probabilities π_0, π_1, $\pi_2, \ldots$ can be computed to any desired precision by the techniques of this section, if only one has sufficient patience and a good calculator.

4.6 Non-Obvious Random-Access Algorithms with Increased Maximum Stable Throughputs

We saw in Section 4.4 that the maximum value of the thoughput λ for which the CORAA is stable is (to three decimal places) .347 packets/slot. For the MCORAA, this "maximum stable throughput" is (again to three decimal places) .375 packets/slot. There are many "non-obvious" ways to devise random-access schemes based on the CCRA (or the MCCAA) to increase the maximum stable throughput. Perhaps the most natural way is to "decouple" transmission times from arrival times, as was first suggested by Gallager [9].

Suppose as before that the random-access scheme is activated at time $t = 0$ and that the unit of time is the slot so that the i-th transmission slot is the time interval $(i,i+1]$. But now suppose that a second time increment Δ has been

chosen to define arrival epochs in the manner that the i-th arrival epoch is the time interval $(i\Delta, i\Delta+\Delta]$. [Note that Δ has units of slots so that $\Delta = 1.5$, for instance, would mean that arrival epochs have length 1.5T seconds, where T is the length of transmission slots in seconds.] Then a very natural way to obtain a random-access algorithm from a collision-resolution algorithm is to use as the first-time transmission rule: transmit a new packet that arrived during the i-th arrival epoch in the first utilizable slot following the collision-resolution interval (CRI) for new packets that arrived during the (i-1)-st arrival epoch. The modifier "utilizable" reflects the fact that the CRI for new packets that arrived during the (i-1)-st arrival epoch may end before the i-th arrival epoch. If so, the CRI for the new packets that arrive during the i-th arrival epoch is begun in the first slot that begins after this arrival epoch ends. The "skipped" transmission slots are wasted, and indeed one could improve the random-access algorithm slightly by "shortening" the i-th arrival epoch in this situation — but this complicates both the analysis and the implementation and has no effect on the maximum stable throughput.

The analytical advantage of the above first-time transmission rule is that it completely eliminates statistical dependencies between the resulting CRI's. If X_i denotes the number of new packets that arrive in the i-th arrival epoch and Y_i denotes the length of the CRI for these packets, then X_0, X_1, X_2,... is a sequence of i.i.d. (independent and identically distributed) random variables, and thus so also is Y_0, Y_1, Y_2,... . Letting X and Y denote an arbitrary pair X_i and Y_i, we note first that, because the new arrival process is Poisson with a mean of λ packets/slot,

$$P(X=N) = \frac{(\lambda\Delta)^N}{N!} e^{-\lambda\Delta}. \tag{4.108}$$

Moreover, we have

$$E(Y) = \sum_{N=0}^{\infty} L_N \, P(X=N) \qquad (4.109)$$

and

$$E(Y^2) = \sum_{N=0}^{\infty} S_N \, P(X=N). \qquad (4.110)$$

In fact, our random-access system is now just a discrete-time queueing system with independent total service times for the arrivals in each arrival epoch. The random-access system is surely unstable if

$$E(Y) > \Delta \qquad (4.111)$$

since then the "server" must fall behind the arrivals. Conversely, if

$$E(Y) < \Delta \qquad (4.112)$$

and $E(Y^2)$ is finite, then the law of large numbers suffices to guarantee that the average waiting time in the queue will be finite and hence that the random-access system will be stable.

Now consider the use of the CCRA with the above first time transmission rule. From (3.21) and Table (3.1), we have

$$L_N \leq aN - 1 + 2\delta_{ON} + (2-a)\delta_{IN} + (6-2a)\delta_{2N} + \left(\frac{26}{3} - 3a\right)\delta_{3N} \qquad (4.113)$$

where

$$a = 2.8867. \qquad (4.114)$$

Substituting (4.113) and (4.108) into (4.109), we find

$$E(Y) \leq f_a(Z) \stackrel{\Delta}{=} aZ - 1 + e^{-Z}[2+(2-a)Z+(3-a)Z^2+\left(\frac{13}{9} - \frac{a}{2}\right)Z^3], \qquad (4.115)$$

where we have defined

$$Z = \lambda\Delta. \qquad (4.116)$$

Using (4.115) in (4.112) [and noting that $E(Y^2)$ is finite as follows from (4.110) and (3.35)], we see that our random-access algorithm will be stable for

$$\lambda < \sup_{Z>0} \frac{Z}{f_a(Z)} = .4294 \qquad (4.117)$$

where the maximizing value of Z is found numerically to be

$$Z = \lambda\Delta = 1.147. \qquad (4.118)$$

This suggests that the maximum throughput is obtained when the length of the arrival epochs is chosen so that the average number of arrivals is 1.147 — however the maximum in (4.117) is very broad, choosing $\lambda\Delta = 1$ gives a system which is stable for $\lambda < .4277$. Moreover, from (3.21) we see that the inequality (4.113) is reversed if in place of (4.114) we take

$$a = 2.8810. \qquad (4.119)$$

The condition (4.111) for <u>instability</u> of the random-access sytstem is then just

$$\lambda > \sup_{Z>0} \frac{Z}{f_a(Z)} = .4295 \qquad (4.120)$$

where the maximum is now attained by $Z = \lambda\Delta = 1.148$. Thus, the <u>maximum stable throughput</u> of this random-access scheme based on the CCRA is (to three decimal places) .429 packets/slot, compared to only .347 packets/slot for the CORAA.

If the above first-time transmission rule is used together with the MCCRA, then an entirely similar argument starting from (3.51) shows that this random-access system is <u>stable</u> for

$$\lambda < .4622$$

(where the maximizing value is $Z = \lambda\Delta = 1.251$) but is <u>unstable</u> for

$$\lambda > .4623.$$

Thus, the maximum stable throughput is (to three decimal places) .462 packets/slot, compared to only .375 packets/slot for the MCORAA.

A little reflection shows that the increased throughput obtained by using the above first-time transmission rule rather than the "obvious" first-time transmission rule, is that the former avoids the very large initial collisions that occur in the latter when the previous CRI has been so long that many new packets are awaiting first-time transmission.

4.7 <u>Other Variations and the Capacity of the Poisson Random-Access Channel</u>

If one defines the <u>Poisson Random-Access Channel</u> by conditions (i),

(ii) and (iii) of Section 2.1 together with the specification that for $t > 0$ the new packet process is a stationary Poisson point process with a mean of λ packets/slot, then a quite reasonable definition of its capacity is as the supremum, over all realizable random-access algorithms, of the maximum stable throughput obtainable with these algorithms. The maximum stable throughput itself is the supremum of those λ for which the average delay experienced by a randomly-chosen packet is finite when the given random-access algorithm is used. It follows from the results of the previous section that the capacity of the Poisson Random-Access Channel is at least .462 packets/slot.

Note that if a random-access algorithm is stable, i.e,, if the average delay for a randomly-chosen packet is finite, then the probability must be unity that a randomly-chosen packet is eventually transmitted successfully. Thus "lossy" random-access algorithms in which there is a nonzero probability that retransmission of a packet is abandoned before it is successfully transmitted are always unstable.

We have specified the branching action within the CCRA and MCCRA to be determined by the results of independent coin flips by the various transmitters concerned. It should be clear that we could equivalently have specified this branching to be determined by the arrival times of the individual packets at their transmitters. For example, if there has been a collision on the first transmission of the packets in some arrival epoch when the CCRA or MCCRA is used together with the first-time transmission rule of the previous section, then "flipping 0" or "flipping 1" by the colliding transmitters is equivalent to "arriving in the first half of the arrival epoch" or "arriving in the second half of the arrival epoch." If all "coin flips" are so implemented by halving the time interval in question, then the resulting random-access algorithm becomes a first-come-first-served (FCFS) algorithm. (The CORAA and MCRAA likewise

become FCFS when this manner of implementing coin flips is used.)

Suppose we use this time-interval-halving method to implement coin flips. As Gallager was first to note [9], if a collision is followed immediately by another collision, then one has obtained no information about the number of packets in the second half of the interval corresponding to the former collision. Thus, the second half interval can be merged into the unexamined portion of the arrival time axis rather than explored as determined by continuation of the collision-resolution algorithm. Using this "trick" with the MCCRA and the first-time transmission rule of the preceding section, Gallager obtained a maximum throughput of .4872 packets/slot (compared to only .462 packets/slot without this "trick.") Mosely [10] refined this approach by optimizing at every step the length of the arrival interval given permission to transmit (which is equivalent to allowing bias in the coin tossed) to obtain a maximum stable throughput of .48785; Mosely also gave quite persuasive arguments that this was optimum for "first-come-first-tried" algorithms.

On the other side of the fence, Pippenger [11] used information-theoretic arguments to show that all realizable algorithms are unstable for $\lambda > .744$, and Humblet [12] sharpened this result to $\lambda > .704$. Very recently, Molle [13] used a "magic genie" argument to show that all realizable algorithms are unstable for $\lambda > .6731$ packets/slot.

Thus, the capacity of the Poisson Random-Access Channel lies somewhere between .48785 packets/slot and .6731 packets/slot, and no more can be said with certainty at this writing. Beginning with Capetanakis [3], most workers on this problem have conjectured that the capacity is 1/2 packet/slot. In any event, it has recently appeared much easier to reduce the upper bound on capacity then to increase the lower bound. And it is no longer defensible for anyone to claim that 1/e is "capacity" in any sense.

5. EFFECTS OF PROPAGATION DELAYS, CHANNEL ERRORS, ETC.

In Section 2.1, we stated the idealized conditions under which the previous analysis of collision-resolution and random-access algorithms was made. We now show how the ideal-case analysis can be easily modified to include more realistic assumptions and also to take advantage of additional information sometimes available in random-access situations.

5.1 Propagation Delays and the Parallel Tree Algorithm

Assumptions (ii) and (iii) in Section 2.1 stipulated that, immediately at the end of each slot, each transmitter received one of 3 possible messages, say "ACK" or "NAK" or "LAK," indicating that one packet had been successfully transmitted in that slot or that there had been a collision in the slot or that the slot had been empty, respectively. This assumption is appropriate, however, only when the round trip transmission time is much smaller than one slot length. Suppose that the round-trip propagation delay time plus the transmission time for the feedback message (ACK or NAK or LAK) is actually D_r slots. (If the propagation delay varies among the transmitters, then D_r is the maximum such delay.) In this case, the result of transmissions in slot i can govern future transmissions no earlier than in slot i + d where

$$d = \lceil D_r \rceil + 1 \tag{5.1}$$

and where $\lceil x \rceil$ denotes the smallest integer equal to or greater than x. For instance, $D_r = 2.3$ slots would imply d = 4 slots.

The simplest way conceptually to extend our former $D_r = 0$ analysis to the case $D_r > 0$ is to treat the actual random-access channel as d interleaved zero-propagation delay channels. Slots 0, d, 2d,... of the actual channel form slots 0, 1, 2,... of the first interleaved channel. Slots 1, d + 1, 2d + 1,... of the actual channel form slots 0, 1, 2,... of the second interleaved channel; etc.

Whatever random-access algorithm is chosen is independently executed on each of the d interleaved channels.

When the CORAA [or the MCORAA] is used on the individual interleaved channels, perhaps the most natural traffic assignment rule for new packets is that, if the new packet arrives at its transmitter during slot $i - 1$, it is assigned to the next occurring interleaved channel, i.e., to the interleaved channel corresponding to slots i, $i + d$, $i + 2d, \ldots$. We now suppose that this assignment rule is adopted. If the expected delay for a randomly-chosen packet for the case $D_r = 0$ and Poisson traffic with a mean of λ packets/slot is $E(D)$, then the expected delay $E(D_i)$ for a randomly-chosen packet for the interleaved scheme and Poisson traffic with the same mean is just

$$E(D_i) = dE\left(D - \frac{1}{2}\right) + \frac{1}{2} . \tag{5.2}$$

This follows from the facts that each interleaved channel still sees Poisson traffic with a mean of λ packets/slot, and that on the average a new packet waits $\frac{1}{2}$ slot before becoming active in the algorithm for the interleaved channel and thus waits $E\left(D - \frac{1}{2}\right)$ further transmission slots on the interleaved channel before the start of its successful transmission. Using (5.2), we can easily convert the delay-throughput characteristic for the $D_r = 0$ case to that for the interleaved scheme.

Despite the naturalness of the above new packet assignment rule for the interleaved CORAA (or the MCORAA), it is obviously inferior to the rule: assign a newly arrived packet randomly to one of the interleaved channels having no collision-resolution interval (CRI) in progress, if any, and otherwise to the next occurring CRI. But it appears difficult to calculate the resulting improvement in performance.

When arrival epochs are distinguished from transmission slots as suggested in Section 4.6, then a good traffic assignment rule for the interleaved channels

is: transmit a new packet that arrived during the i-th arrival epoch in the next slot that occurs after new packets in the (i-1)-st arrival epoch have been initially transmitted and is from an interleaved channel with no CRI in progress.

In the infinitely-many-sources-generating-Poisson-traffic model, there is never a queue at the individual transmitters. In the practical case, however, a transmitter may receive one or more additional new packets before successfully transmitting a given new packet. In the interleaved case, this circumstance can be exploited to reduce the expected delay by assigning the additional new packets to other interleaved channels so that one transmitter can be actively processing several packets at once. Of course, this means also that these packets might be successfully transmitted in different order from their initial arrivals.

The interleaved scheme for coping with propagation delay is probably the simplest to implement as well as to analyze. It does, however, achieve this simplicity at the price of some increased delay. An alternative to interleaving suggested by Capetanakis [3,4] is to reorder the transmissions in the CCRA so as to process all the nodes at each level in the tree before going on to the next level. For instance, for the situation illustrated in Figures 2.1 and 2.2, the order of the slots would effectively be permuted to 1,2,5,3,4,6, 7,8,11,9,10. Note for instance that the transmitters colliding in slot "5" are idle for the following two slots so that if $d = 1$ or $d = 2$ there would be no wait required before they could proceed with the algorithm. Capetanakis called this scheme the parallel tree algorithm to contrast it with the "serial tree algorithm" (or CCRA in our terminology.)

The parallel tree algorithm appears attractive for use in random-access systems where the propagation delays are large. Thus, it seems worthwhile to

show here that it can be implemented just as easily as the CCRA. Just as was done in Section 2.3 when considering the CCRA, we suppose that each transmitter keeps two counters (which we now call C_1 and C_2), the first of which indicates by a zero reading that he should transmit and the second of which indicates the number of additional slots that have already been allocated for further transmissions within the CRI in progress. The parallel tree algorithm differs from the CCRA only in the respect that, after a collision, the colliding transmitters go to the end of the waiting line rather than remaining at the front. Thus, with the stipulation that all transmitters set C_2 to 1 just before the CRI begins, we can implement the parallel tree algorithm by the following modification of the implementation given for the CCRA in Section 2.3.

Parallel Tree Algorithm Implementation: When a transmitter flips 0 or 1 following a collision in which he is involved, he sets counter C_1 equal to C_2-1 or C_2, respectively, then increments C_2 by 1. After all other slots, he decrements C_1 by 1, and transmits in the next slot when C_1 reaches 0. After non-collision slots, he decrements C_2 by 1; after collisions in which he was not involved, he increments C_2 by 1. When C_2 reaches 0, the CRI is complete.

Notice that, after a collision, those colliding transmitters who flip 0 will transmit again exactly C_2 slots later, i.e., in slot $i + C_2$ when the collision was in slot i. Thus, the following artifice suffices to ensure that the ACK, NAK or LAK reply from the receiver about some slot will reach every transmitter before it is called upon to act on the result of transmissions in that slot.

Parallel Tree Algorithm Implementation When Propagation Delays are Large and Interleaving is Not Used: Same as the ordinary implementation given above except that, when a collision occurs with $C_2 < d$, all transmitters immediately reset C_2 to d before proceeding with the other actions required by the algorithm.

We note that, when C_2 is reset to d, d - C_2 slots will be "wasted" by the algorithm, i.e., these slots will necessarily contain no transmissions. These slots could of course be used to begin (or to continue with) another collision resolution interval — but this would so complicate the implementation that one would probably be better advised to use the interleaved approach if the efficient use of all channel slots is so important as to warrant such complexity.

5.2 Effects of Channel Errors

Assumptions (i) and (ii) of Section 2.1 stipulated that, except for collisions in the forward channel, both the forward and feedback channels in the random-access model were "noiseless." In other words, after each transmission slot, all transmitters are correctly informed as to whether the slot was empty ("LAK"), or contained one packet ("ACK"), or contained a collision ("NAK"). We now consider the more realistic situation where channel noise can affect the transmissions on the forward channel or feedback channel, or both.

The basic model for our error analysis is the discrete memoryless channel shown in Figure 5.1. The input to this channel is the actual status of the given transmission slot, i.e., "blank" or containing a "single packet" or containing a "collision." The output is the actual feedback message reaching the transmitters, i.e., "LAK" or "ACK" or "NAK." In practice, each packet transmitted would be encoded with a sufficiently powerful error-detecting code that the probability would be negligible that the common receiver would incorrectly identify either a "blank" or a "collision" as a "single packet" because of errors in the forward channel. We also presume that the transmitters will interpret any message garbled by noise on the feedback channel as a "NAK," and that these feedback messages would also be coded to make negligible the probability that the transmitters would incorrectly identify either a feedback "LAK" or a "NAK" as an "ACK," or would incorrectly identify either a feedback

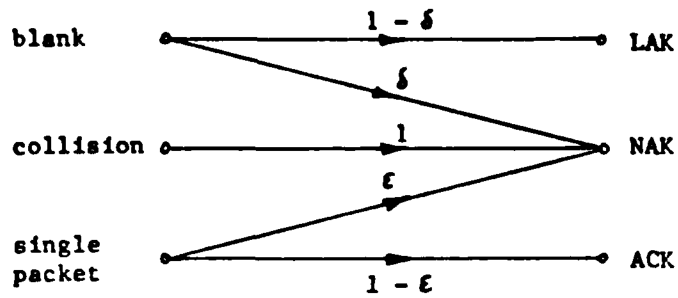


Fig. 5.1:Model for analysis of channel error effects.

"ACK" or "NAK" as an "ACK," because of errors on the feedback channel. Thus, it is realistic to assume that the only types of errors that can occur in our random-access system are those that result in the transmitters interpreting the feedback message as "NAK" when in fact the slot had been "blank" or had contained a "collision." These two types of errors are indicated by the two non-direct transitions shown in the error model of Figure 5.1. We write δ to denote the probability that the transmitters will incorrectly conclude that a blank transmission slot contained a collision, and ε to denote the probability that they will incorrectly conclude that a transmission slot with a single packet contained a collision. Note that δ (as well as ε) accounts for error effects on both the forward and feedback channels. A blank slot could, because of errors on the forward channel, reach the common receiver as a "garble," thus eliciting a "NAK" message on the feedback channel; or a blank slot could be correctly identified by the common receiver but the subsequent "ACK" message, because of noise on the feedback channel, might reach the transmitters as a "garble" that they are forced to interpret as a "NAK" — both of these cases

are included in the transition from "blank" to "NAK" for the error model of Figure 5.1.

In a realistic random-access situation, one would expect δ to be much smaller than ε, as only one bit error in a packet on the forward channel could cause a "single packet" slot to result in a "NAK" by the common receiver (unless some error-correction were employed in addition to error detection). Moreover, for the system to be useful, one requires $\varepsilon << 1$, for otherwise the throughput would be low because of the need for transmitting a packet many times before it is successfully received even if a "genie" were to assist the transmitters to schedule their transmissions so that collisions never occurred. Thus, typically, one would anticipate the inequality

$$\delta << \varepsilon << 1. \tag{5.3}$$

However, we need not impose this requirement in order to analyze the effect of errors on the Capetanakis collision-resolution algorithm (CCRA).

We begin our analysis somewhat indirectly by first extending our analysis of the CCRA for the error-free case. Recalling the definitions of Section 3.1, we now further define Y_b to be the number of blank slots in the CRI, Y_s to be the number of slots with a single packet, and Y_c to be the number of slots with collisions. Referring to the tree diagram (such as in Figure 2.2) for the CRI, we see that $Y_b + Y_s$ is the number of terminal nodes whereas Y_c is the number of intermediate nodes. But, as we noted in Section 2.3, a binary rooted tree always has exactly one more terminal node than intermediate nodes so that

$$Y_b + Y_s = Y_c + 1 \tag{5.4}$$

We next define

$$B_N = E(Y_b|X=N) \tag{5.5}$$

and

$$C_N = E(Y_c|X=N). \tag{5.6}$$

Now we observe that

$$E(Y_s | X=N) = N \tag{5.7}$$

since each of the N packets in the first slot of the CRI is successfully transmitted exactly once in the CRI. But, of course

$$Y = Y_b + Y_s + Y_c \tag{5.8}$$

so that, upon taking the expectation conditioned on X = N, we have

$$L_N = B_N + N + C_N. \tag{5.9}$$

Similarly, (5.4) yields

$$B_N + N = C_N + 1. \tag{5.10}$$

Solving (5.9) and (5.10), we obtain

$$B_N = (L_N+1-2N)/2 \tag{5.11}$$

which is a fundamental relationship for the CCRA in the error-free case.

We are now ready to grapple with errors. In Figure 5.2, we show the effect in the resulting tree diagram for the CRI of a blank-to-NAK and a single-packet-to-NAK error on the operation of the CCRA. Here, a cross inside a node indicates that, because of channel errors, that node will be interpreted as a collision slot by the transmitters. The question mark above the subsequent nodes indicates that we do not yet know whether, because of further channel

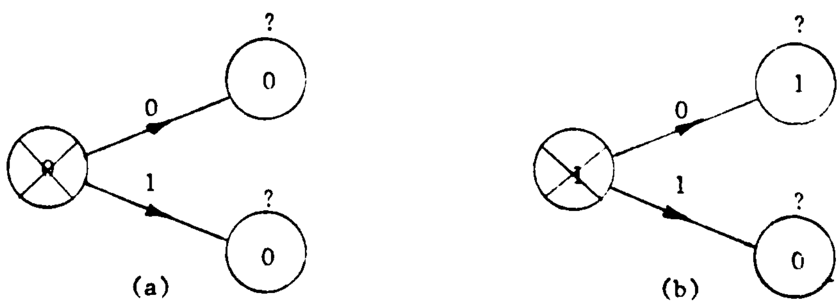


Fig. 5.2: Immediate effect of (a) a blank-to-NAK error and (b) a single-packet-to-NAK error in the CCRA.

errors, that node will also be interpreted as a collision slot rather than correctly as a blank slot or single packet slot. Thus, a blank-to-NAK error has the immediate effect of adding two additional blank slots to the CRI, whereas a single-packet-to-NAK error adds one blank slot and one single packet slot.

We now define the random variable t_b to be the number of slots required for the successful transmission of a given blank slot when errors are present, i.e., the number of slots until all spurious "collisions" have been resolved and found to have been actually blank slots. From Figure 5.2(a), we see that

$$E(t_b) = 1(1-\delta) + [1+2E(t_b)]\delta, \tag{5.12}$$

because $t_b = 1$ when the blank slot initially results in a correct LAK reply as happens with probability $1 - \delta$, but on the average $2E(t_b)$ further slots will be required if the blank slot initially results in an erroneous NAK reply as happens with probability δ. From (5.12) we find

$$E(t_b) - 1 = \frac{2\delta}{1-2\delta} \quad , \quad \delta < \frac{1}{2} \tag{5.13}$$

where we have focused interest on $E(t_b) - 1$ which is the <u>expected number of extra slots added because of channel errors to the CRI for each blank slot in the error-free case</u>.

Similarly, we define t_s to be the number of slots required for the successful transmission of a given single packet slot when errors are present. Referring to Figure 5.2(b), we see that

$$E(t_s) = 1(1-\varepsilon) + [1+E(t_b)+E(t_s)]\varepsilon. \tag{5.14}$$

Substituting (5.13) into (5.14) and solving, we find

$$E(t_s) - 1 = \frac{2\varepsilon(1-\delta)}{(1-2\delta)(1-\varepsilon)} \quad , \quad \delta < \frac{1}{2} , \ \varepsilon < 1 \tag{5.15}$$

which is the <u>expected number of extra slots added because of channel errors to the CRI for each single-packet slot in the error-free case</u>.

It now follows immediately from (5.13) and (5.15) that

$$E(Y|X=N,\text{errors}) = L_N + B_N \frac{2\delta}{1-2\delta} + N \frac{2\varepsilon(1-\delta)}{(1-2\delta)(1-\varepsilon)} . \quad (5.16)$$

Using (5.11) and (5.16) and simplifying, we find

$$E(Y|X=N,\text{errors}) = \frac{1-\delta}{1-2\delta} L_N + \frac{2(\varepsilon-\delta)}{(1-2\delta)(1-\varepsilon)} N + \frac{\delta}{1-2\delta} . \quad (5.17)$$

The fundamental relation (5.17) now permits us to make use of the tight bounds on L_N developed in Section 3.3 to obtain tight bounds on the expected CRI length in the presence of errors.

In particular, it follows from (3.21) and (5.17) that

$$E(Y|X=N,\text{errors}) \le \left[\frac{2.8867(1-\delta)}{1-2\delta} + \frac{2(\varepsilon-\delta)}{(1-2\delta)(1-\varepsilon)}\right] N - 1 \quad (5.18)$$

for all $N \geq 4$. The stability analysis of Sections 4.3 and 4.4 can now immediately be invoked to assert that <u>the CORAA is stable in the presence of channel errors provided that</u>

$$\lambda < \left[\frac{2.8867(1-\delta)}{1-2\delta} + \frac{2(\varepsilon-\delta)}{(1-2\delta)(1-\varepsilon)}\right]^{-1} \quad (5.19)$$

where λ is the throughput, i.e., the average number of new packets per slot, of the Poisson traffic. Moreover, we know from the tightness of the upper bound (3.21) on L_N that the right side of (5.19) is very nearly equal to the maximum stable throughput.

In Table 5.1, we show the tight lower bound on the maximum stable throughput given by the right side of (5.19) over a wide range of ε and δ. The values of ε and δ have not been chosen to correspond to a practical situation [cf. (5.3)], but rather to demonstrate that <u>the CORAA is remarkably insensitive to channel errors</u>. Even for the practically extreme values $\varepsilon = \delta = .1$, the maximum stable throughput is still 90% of its value in the error-free case.

In light of the insensitivity of the CORAA to channel errors, it seems quite surprising that the MCORAA is extremely sensitive to such errors, as we now proceed to show. We will say that a collision-resolution algorithm suffers

ε	δ	Lower Bound (5.19) on Maximum Stable Throughput
0	0	.3464
.01	0	.3440
.10	0	.3217
.20	0	.2953
.50	0	.2046
.80	0	.0919
0	.01	.3453
0	.10	.3336
0	.20	.3142
0	.40	.2146
0	.45	.1454
.10	.10	.3079
.20	.20	.2598
.30	.30	.1980
.40	.40	.1155
.10	.01	.3205
.10	.02	.3193
.20	.01	.2940
.20	.02	.2928

Table 5.1: The Lower Bound (5.19) on the Maximum Stable Throughput of the CORAA for Various Values of ε (the single-packet-to-NAK error probability) and δ (the blank-to-NAK error probability).

from deadlock due to channel errors, if, because of a finite number of errors, the resulting CRI never terminates although its first slot contains only a finite number of packets. Clearly, a collosion-resolution algorithm with such

deadlock would be a disastrous choice for inclusion within a random-access algorithm.

In Figure 5.3, we give an example to show that the modified Capetanakis collision-resolution algorithm (MCCRA) suffers from deadlock due to channel errors. In this example, the first slot of the CRI is actually blank but,

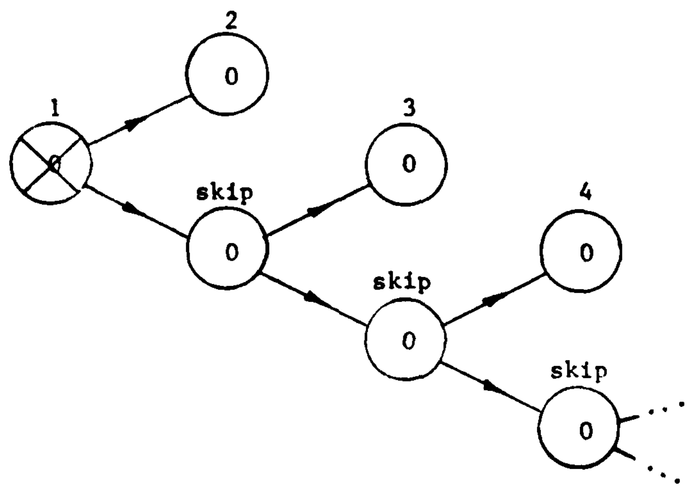


Fig. 5.3: Example of deadlock due to channel errors in the MCCRA.

because of a blank-to-NAK error, is construed by the transmitters to have contained a collision. They all wait for those colliding transmitters who flip 0 to send in slot 2, which of course must then be blank. But the MCCRA now directs the colliding transmitters to skip what erroneously appears to be a certain collision among the colliding transmitters who flipped one. These latter are directed to flip again with those who now flip 0 then transmitting in slot 3, which of course must again be blank, etc. A single blank-to-NAK error thus results in an infinitely long CRI even though there are actually no packets to be transmitted!

Naturally one could "doctor" the MCCRA to avoid the above illustrated

deadlock, say by specifying that no more than 2 successive blank-skip slots will be permitted. But this further complicates its implementation and also reduces its maximum stable throughput under error-free conditions. When one reflects that the maximum stable throughputs of the CORAA and the MCORAA are .347 and .375, respectively [and that, for the non-obvious first time transmission rule of Section 4.6, are .430 and .462 for the CCRA and MCCRA, respectively] in the error-free case, one can hardly escape the conclusion that the slightly increased throughput of the MCCRA comes at too great a price in increased sensitivity to channel errors compared to the CCRA.

Returning to consideration of the CCRA, we now address some possible objections to our error analysis. One might argue that we should have allowed collision-to-ACK or collision-to-NAK errors in our analysis, even though they may have very small probabilities. But a collision-to-ACK error will cause all the colliding transmitters to believe that their packets were successfully transmitted and thus will actually shorten the CRI. Naturally these packets are forever lost, unless some accounting is performed at the destination and a repeat-request sent back — but this is the usual and unavoidable way that any communication system must occasionally fail, and is not related to random-access issues. On the other hand, a collision-to-LAK error will be immediately recognized by all the colliding transmitters who then presumably would save their packets for transmission in the next CRI. The only two conceivable types of errors that remain unconsidered are blank-to-ACK errors, which cause no disruption at all in the CCRA, and single-packet-to-LAK errors, which again will each be recognized by the transmitter in question so that the packet can be transmitted in the next CRI. The heartening conclusion is that <u>the CCRA is extermely insensitive to channel errors of every possible sort</u>.

Finally, one could object that we have not allowed the possibility that a

feedback message from the common receiver could be correctly received by some transmitters but incorrectly understood as a NAK by the rest. It is tedious, but not difficult, to enumerate all types of such errors and to find again that none does significant damage to the operation of the CCRA. The robustness of the CCRA in the face of every sort of error really stems from the fact that a priori it assumes always that there may be a collision in the next slot. If an unauthorized packet appears in this slot because of channel errors on past transmissions, it merely becomes a part of the collision to be resolved from that point onwards. This packet's shady past does not preclude a bright future. The CCRA even rides through periods where some transmitters lose track of the correct endpoints of the CRI's because eventually from this point onwards there will be more non-collision slots than collision slots so that all transmitters will again agree on the endpoint of a CRI. The MCCRA, however, is prone to disaster from channel errors because it rashly reaches the conclusion that a collision is certain to occur in some slot from imperfect information garnered from previous slots.

5.3 Carrier-Sensing

In some random-access situations, particularly in packet-radio networks, it is possible for the transmitters to "hear" that a transmission slot is empty or, when sending a packet, to "hear" that interfering signals are present. In either case, the transmitters can then abruptly terminate the otherwise unproductive slot. Such techniques are generally known as "carrier-sensing" [14], and we now analyze their effectiveness in conjunction with the Capetanakis collision-resolution algorithm (CCRA).

Let θ_b be the fraction of a slot required for all transmitters to detect that a transmission slot is empty (i.e., "blank"), and let θ_c similarly be the fraction of a slot required to detect that a transmission slot contains a

collision. Then, if carrier-sensing (c-s) is exploited to terminate blank and collision slots as soon as they are detected by all transmitters, the effect is merely to reduce the length of each blank or collision slot by a factor of $1 - \theta_b$ or $1 - \theta_c$, respectively. Thus, it follows immediately that the expected length of a collision-resolution interval (CRI) for the CCRA is given by

$$E(Y|X=N,c\text{-}s) = L_N - (1-\theta_b)B_N - (1-\theta_c)C_N \tag{5.20}$$

where B_N and C_N are as defined in the previous section.

Solving (5.9) and (5.10), we find

$$C_N = (L_N-1)/2. \tag{5.21}$$

Now using (5.11) and (5.21) in (5.20), we obtain

$$E(Y|X=N,c\text{-}s) = L_N(\theta_b+\theta_c)/2 + N(1-\theta_b) + (\theta_b-\theta_c)/2, \tag{5.22}$$

which is a fundamental relation. Just as for the error analysis of the preceding section, we can now exploit in (5.22) the tight bounds on L_N developed in Section 3.3.

Using (3.21) in (5.22), we find

$$E(Y|X=N,c\text{-}s) \leq [1.4434(\theta_b+\theta_c) + 1-\theta_b]N - \theta_c \tag{5.23}$$

for all $N \geq 4$. The stability analysis of Sections 4.3 and 4.4 can now immediately be invoked to assert that the CORAA is stable provided that

$$\lambda < [1.4434(\theta_b+\theta_c) + 1 - \theta_b]^{-1}, \tag{5.24}$$

where λ is the throughput, i.e., the average number of new packets per (unshortened) slot, of the Poisson traffic. Again we know from the tightness of the upper bound (3.21) on L_N that the right side of (5.24) is virtually equal to the maximum stable throughput.

In Table 5.2, we show the tight lower bound on the maximum stable throughput of the CORAA given by the right side of (5.24) over a wide range of θ_b and

θ_c. Not surprisingly, the maximum stable throughput is unity for $\theta_b = \theta_c = 0$, since then blank slots and collision slots are reduced to zero length so that every actual transmission slot can be used for successful transmissions. More significantly, Table 5.2 shows that early detection of collisions is far more important than early detection of blank slots. For instance, the maximum stable

θ_b	θ_c	Lower Bound (5.24) on Maximum Stable Throughput
0	0	1
.1	.1	.841
.2	.2	.726
.5	.5	.515
.7	.7	.431
1	1	.346
1	.9	.365
1	.7	.408
1	.5	.462
1	.2	.577
1	0	.693
.5	1	.375
.2	1	.395
0	1	.409

Table 5.2: The Lower Bound (5.24) on the Maximum Stable Throughput of the CORAA for Various Values of θ_b (fraction of slot required to detect no transmission) and θ_c (fraction of slot required to detect a collision).

throughput for instant detection of blank slots (θ_b=0) is only .409 when $\theta_c = 1$, compared to .346 when $\theta_b = \theta_c = 1$; but this same increased stable throughput can

be attained by the rather late detection of collisions, $\theta_c = .7$, when $\theta_b = 1$.

5.4 Etc.

Virtually any of the "tricks," such as making short "reservations" on the channel for the later transmission of long "messages" [15], that have been suggested for increasing the throughput of Aloha-like random-access systems can also be incorporated into random-access systems based on the Capetanakis collision-resolution algorithm (CCRA). Our aim in the previous section has been to illustrate for one such "trick," viz., carrier-sensing, how our previous analysis of the CCRA and the CORAA can readily be extended to calculate the resulting enhancement of the random-access system without any appeal to "statistical equilibrium." The reader should have no difficulty in making the appropriate extensions for the other tricks in the bag of the random-access system designer.

6. SUMMARY AND HISTORICAL CREDITS

In the previous text, we have given a rather thorough analysis of the Capetanakis collision-resolution algorithm (CCRA) and its use within random-access systems. We have stressed the algorithmic properties of the CCRA itself so that our main results are independent of traffic statistics. When calculating the performance of random-access systems based on the CCRA, however, we have generally used the usual Poisson traffic model. As we have repeatedly emphasized, all of our calculations have been mathematically rigorous; in particular, they have been devoid of evasive appeals to the assumption of "statistical equilibrium" that has been all too pervasive in the random-access literature. Finally, we have extended our analysis to include the effects of channel errors and propagation delays, and to calculate the benefits from "carrier-sensing."

In many places in the text, we have pointed out the original source of the result being discussed, but we now attempt to fill in as many omissions of such credit as possible.

The pioneering work of Capetanakis [3-5] has of course been the main source and inspiration of this paper. It is doubtful that any scheme so elegantly simple as the CCRA algorithm has no prior roots in the literature. Some such roots of the CCRA can be seen in the polling algorithm proposed by Hayes [16], but the fundamental concept of a collision-resolution algorithm seems to have found no expression prior to the work of Capetanakis.

The author suggested to Capetanakis the modification to eliminate certain collisions that we have called the MCCRA — not knowing at the time that it leads to deadlock with channel errors. Quite interestingly, this MCCRA was independently but somewhat later discovered by Tsybakov and Mihailov [7], who bypassed the more robust algorithm. These latter authors, however, were the first to use a recursive analysis in their study of the MCCRA and MCORAA. The

extremely tight and systematic bounds of Section 3 are due to the author's student, M. Amati, and will form part of his doctoral dissertation. The analysis of Sections 4.1 through 4.5 is largely the joint work of Amati and the author, but the maximum stable throughputs calculated there had already been found by Capetanakis [3,4] using Chernoff bounds.

Capetanakis [3,4] originally gave a modification of his algorithms, based on dynamically varying the degree of the root node in the tree, that yields the same maximum stable throughputs of .429 and .462 as calculated in Section 4.6; our approach there, which was based on Gallager's trick of divorcing the arrival time axis from the transmission time axis, is virtually equivalent to, but conceptually simpler than, Capetanakis' "dynamic tree algorithm." The implementation of the parallel tree algorithm suggested in Section 5.1 is due to the author, as is the analysis of Sections 5.2 and 5.3, and as are any errors that the reader may have found.

ACKNOWLEDGEMENT

The support by the U.S. Office of Naval Research (Contract No. N00014-78-C-0778) of the original research by M. Amati and the author reported herein is gratefully acknowledged.

REFERENCES

[1] N. Abramson, "The ALOHA system — another alternative for computer communications," AFIPS Conf. Proc., Fall Joint Comp. Conf., vol. 37, pp. 281-285, 1970.

[2] L. G. Roberts, "ALOHA packet system with and without slots and capture," ASS Note 8, ARPA Network Info. Ctr., Stanford Res. Inst., Stanford, CA, June 1972.

[3] J. I. Capetanakis, "The multiple access broadcast channel: protocol and capacity considerations," Ph.D. Thesis, Course VI, Mass. Inst. Tech., Cambridge, MA, August 1977; orally presented at IEEE Int. Symp. Info. Th., Ithaca, NY, October 10-14, 1977.

[4] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," IEEE Trans. Info. Th., vol. IT-25, pp. 505-515, September 1979.

[5] J. I. Capetanakis, "Generalized TDMA: The multi-accessing tree protocol," IEEE Trans. Comm., vol. COM-27, pp. 1476-1484, October 1979.

[6] R. G. Gallager, Private Communication, January 1978.

[7] B. S. Tsybakov and V. A. Mihailov, "Slotted multiaccess packet-broadcasting feedback channel," Problemy Peredachi Informatsii, vol. 14, pp. 32-59, October-December 1978.

[8] A. J. Viterbi and J. K. Omura, Principles of Digital Communication and Coding, New York, McGraw-Hill, 1979, pp. 40-42.

[9] R. G. Gallager, "Conflict resolution in random access broadcast networks," preliminary manuscript, 1979.

[10] J. Mosely, "An efficient contention resolution algorithm for multiple access channels," Tech. Rpt. LIDS-TH-918, Lab. Info. Dec. Sys., Mass. Inst. of Tech., Cambridge, MA, June 1979.

[11] N. Pippenger, "Bounds on the performance of protocols for a multi-access broadcast channel," Res. Rpt. 7742 (#33525), IBM Res. Ctr., Yorktown Heights, NY, June 28, 1979.

[12] P. Humblet, Private Communication, November 1979.

[13] M. Molle, "On the capacity of infinite population multiple access protocols," submitted to IEEE Trans. Info. Th., April 1980.

[14] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I — Carrier sense multiple-access modes and their throughput-delay characteristics," IEEE Trans. Comm., vol. COM-23, pp. 1400-1416, December 1975.

[15] L. G. Roberts, "Dynamic allocation of satellite capacity through packet reservations," Proc. Nat. Comp. Conf., pp. 711-716, 1973.

[16] J. F. Hayes, "An adaptive technique for local distribution," IEEE Trans. Comm., vol. COM-26, pp. 1178-1186, August 1978.

DATE

FILMED

8-8